ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка методики побудови хмарного сховища на базі сучасних методів захисту даних»

на здобуття освітнього ступеня магістра
зі спеціальності 121 Інженерія програмного забезпечення
(код, найменування спеціальності)
освітньо-професійної програми «Інженерія програмного забезпечення»
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Андрій МИСЛЮК
(підпис)

Виконав: здобувач вищої освіти групи ПДМ-64
Андрій МИСЛЮК

Керівник: Андрій БОНДАРЧУК
д.т.н., професор

Рецензент: _____
науковий ступінь, вчене звання *Ім'я, ПРІЗВИЩЕ*

Київ 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Інженерії програмного забезпечення
_____ Ірина ЗАМРІЙ
«_____» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Мислюку Андрію Сергійовичу

1. Тема кваліфікаційної роботи: «Розробка методики побудови хмарного сховища на базі сучасних методів захисту даних»

керівник кваліфікаційної роботи Андрій БОНДАРЧУК д.т.н., професор,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» жовтня 2023 р. №145.

2. Строк подання кваліфікаційної роботи «29» грудня 2023 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, дослідження існуючих хмарних сховищ, методи захисту даних у хмарних сховищах, технічні параметри та вимоги до хмарних сховищ.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз архітектури хмарного сховища.
2. Дослідження процедури реєстрації та автентифікації користувачів.
3. Розробка нових стратегій шифрування та методів захисту даних.
4. Оцінка масштабованості та надійності системи.

5. Впровадження теоретичних концепцій у реальному демонстраційному додатку.

5. Перелік графічного матеріалу: *презентація*
   1. Порівняння існуючих хмарних сховищ;
   2. Архітектурна діаграма хмарного сховища;
   3. Алгоритм процесу авторизації користувача;
   4. Алгоритм процесу керування сесіями користувача;
   5. Алгоритм процесу отримання даних по захищеному каналу;
   6. Приклад керування активними сесіями користувачів;
   7. Приклад відправки повідомлень через centrifuge;
   8. Математичні формули для оцінки ефективності системи;
   9. Оцінка масштабованості та надійності системи.

6. Дата видачі завдання «19» жовтня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	19.10-05.11.23	
2	Вивчення матеріалів для архітектури хмарного сховища	06.11-12.11.23	
3	Дослідження процедури реєстрації та автентифікації користувачів	13.11-19.11.23	
4	Дослідження захисту даних та їх шифрування	20.11-26.11.23	
5	Аналіз процесів масштабованості та надійності системи	27.11-03.12.23	
6	Застосування теоретичних концепцій у практичному демонстраційному додатку	04.12-10.12.23	
7	Оформлення роботи: вступ, висновки, реферат	11.12-20.12.23	
8	Розробка демонстраційних матеріалів	21.12-29.12.23	

Здобувач вищої освіти _____ Андрій МИСЛЮК
(підпис)

Керівник
кваліфікаційної роботи _____ Андрій БОНДАРЧУК
(підпис)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 65 стор., 1 табл., 17 рис., 26 джерел.

Мета роботи – розробка методики побудови хмарного сховища, яка інтегрує сучасні методи захисту даних, з акцентом на розробці ефективних систем авторизації та контролю сесій.

Об'єкт дослідження – побудова та вдосконалення хмарних сховищ з використанням сучасних методів захисту даних.

Предмет дослідження – методи та засоби для створення хмарного сховища, що включає інноваційні підходи в захисті даних і механізми авторизації та контролю сесій.

Короткий зміст роботи: Робота охоплює аналіз існуючих хмарних сховищ, розробку нових підходів до архітектури хмарних сховищ, дослідження процедур реєстрації та автентифікації користувачів, а також методів захисту та шифрування даних. Також здійснюється оцінка масштабованості та надійності системи, з акцентом на застосування теоретичних концепцій у реальному демонстраційному додатку.

КЛЮЧОВІ СЛОВА: ХМАРНІ СХОВИЩА, ЗАХИСТ ДАНИХ, ШИФРУВАННЯ, АРХІТЕКТУРА ХМАРНИХ СИСТЕМ, МАСШТАБОВАНІСТЬ, НАДІЙНІСТЬ.

ABSTRACT

Text part of the master's qualification work:
65 pages, 17 pictures, 1 table, 26 sources.

The goal of this work is to develop a methodology for building a cloud storage system that integrates modern data protection methods, with a focus on developing effective systems for authorization and session control.

Object of research – the construction and improvement of cloud storages using modern data protection methods.

Subject of research – methods and tools for creating a cloud storage that includes innovative approaches in data protection and mechanisms for authorization and session control.

Summary of the work: The work encompasses the analysis of existing cloud storages, the development of new approaches to cloud storage architecture, the study of user registration and authentication procedures, as well as data protection and encryption methods. It also involves evaluating the scalability and reliability of the system, with an emphasis on applying theoretical concepts in a real demonstration application.

KEYWORDS: CLOUD STORAGE, DATA PROTECTION, ENCRYPTION, CLOUD SYSTEM ARCHITECTURE, SCALABILITY, RELIABILITY.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface

IT – Information Technology

IaaS – Infrastructure as a Service

PaaS – Platform as a Service

GDPR – General Data Protection Regulation

HIPAA – Health Insurance Portability and Accountability Act

CORS – Cross-Origin Resource Sharing

AES – Advanced Encryption Standard

RSA – Rivest–Shamir–Adleman

TLS – Transport Layer Security

SSL – Secure Sockets Layer

ECC – Elliptic Curve Cryptography

SHA-256 – Secure Hash Algorithm 256-bit

AES – Advanced Encryption Standard

PGP – Pretty Good Privacy

HSM – Hardware Security Module

TDE – Transparent Data Encryption

OAuth – Open Authorization

SAML – Security Assertion Markup Language

2FA – Two-Factor Authentication

WAF – Web Application Firewall

VPN – Virtual Private Network

IDS – Intrusion Detection System

KMS – Key Management Service

MFA – Multi-Factor Authentication

FDE – Full Disk Encryption

IPsec – Internet Protocol Security

ВСТУП

У сучасному цифровому світі, хмарні технології відіграють ключову роль у зберіганні та обробці величезних обсягів даних. Зростаюча залежність від хмарних сервісів вимагає постійного вдосконалення методів захисту даних. В цьому контексті, розробка надійних хмарних сховищ, здатних забезпечити високий рівень безпеки та приватності, стає надзвичайно важливою.

Актуальність дослідження. Хмарні технології стають все більш важливими для зберігання та обробки даних, актуальність дослідження ефективних методів захисту даних у хмарних сховищах набуває особливого значення. Зростаюча залежність від хмарних сервісів та необхідність забезпечення високого рівня безпеки та конфіденційності ставлять перед дослідниками нові виклики.

Мета роботи – розробка методики побудови хмарного сховища, яка інтегрує сучасні методи захисту даних, з акцентом на розробці ефективних систем авторизації та контролю сесій.

Об'єкт дослідження – побудова та вдосконалення хмарних сховищ з використанням сучасних методів захисту даних.

Предмет дослідження – методи та засоби для створення хмарного сховища, що включає інноваційні підходи в захисті даних і механізми авторизації та контролю сесій. Особлива увага приділяється аналізу масштабованості та надійності хмарних систем, а також вивченню сучасних тенденцій у розвитку хмарних технологій.

Завдання дослідження:

- аналіз архітектури хмарного сховища;
- дослідження процедури реєстрації та автентифікації користувачів;
- розробка нових стратегій шифрування та методів захисту даних;
- оцінка масштабованості та надійності системи;
- впровадження теоретичних концепцій у реальному демонстраційному додатку.

Методи дослідження. Дослідження передбачає використання аналітичних методів для вивчення існуючих хмарних технологій та методів захисту даних, експериментальних методів для розробки та тестування нових рішень, а також методів системного аналізу для оцінки ефективності та масштабованості хмарних систем.

Для досягнення визначеної мети, робота передбачає детальне вивчення сучасних технологій шифрування та безпеки даних, аналіз ефективності існуючих методів аутентифікації, а також розробку нових рішень для підвищення безпеки хмарних систем. На увагу також варто звернути на впровадження новітніх технологічних інновацій та адаптацію існуючих рішень до вимог сучасного цифрового середовища. Особлива увага приділяється інтеграції автоматизації процесів безпеки та оптимізації управління хмарними ресурсами.

Узагальнюючи, метою є розробка та валідація інноваційних підходів до захисту хмарних сховищ, які враховують сучасні технологічні можливості та зростаючі вимоги до конфіденційності та безпеки в цифровому світі. Результати дослідження пропонують нові перспективи для підвищення ефективності хмарних систем та забезпечення високого рівня захисту даних в широкому спектрі застосувань.

1 ОГЛЯД І АНАЛІЗ ХМАРНИХ СХОВИЩ

1.1 Архітектура хмарних сховищ

Сьогодні хмарні сховища відіграють ключову роль у цифровій екосистемі, надаючи можливості для зберігання, обробки та доступу до величезних обсягів даних. Хмарне сховище – це система, що дозволяє зберігати дані на віддалених серверах, доступних через Інтернет, і яка базується на моделі хмарних обчислень. Цей сектор швидко розвивається, пропонуючи новітні технології та підходи до зберігання та захисту даних [1].

Архітектура хмарних сховищ охоплює різноманітні компоненти та підсистеми, її частини є складною системою, яка включає ряд ключових елементів, кожен з яких відіграє важливу роль у забезпеченні ефективного зберігання та обробки даних.

На початку є клієнтські програми. Клієнтські програми – це інтерфейси, через які користувачі взаємодіють з хмарним сховищем [2]. Вони можуть бути реалізовані як десктопні додатки, мобільні застосунки або веб-інтерфейси. Важливість цих програм полягає в тому, що вони надають простий і зрозумілий спосіб завантаження, доступу та управління файлами у хмарі. Вони також мають забезпечувати безпеку, наприклад, через аутентифікацію та шифрування.

Далі йдуть сервери даних, які є основою хмарного сховища. Сервери даних – це потужні комп'ютери або серверні кластери, які фізично зберігають дані користувачів у хмарних сховищах. Ці сервери, зазвичай розташовані в дата-центрах, можуть зберігати великі обсяги інформації і забезпечують її доступність з будь-якого місця в світі. Вони також повинні бути надійними та масштабованими, щоб впоратися з великою кількістю запитів та забезпечити високу швидкість передачі даних [3].

Системи управління даними відповідають за організацію та обробку даних у хмарі. Системи управління даними - це системи, які складаються з різноманітних

інструментів та технологій, які використовуються для управління, організації та структурування даних у хмарних сховищах [4]. Вони включають бази даних та файлові системи, які забезпечують структуру та ефективний доступ до інформації. Ці системи також включають інструменти для аналізу даних, пошуку та керування доступом, що дозволяє користувачам легко знаходити та використовувати потрібні їм дані.

Що стосується шифрування та безпеки, то цей елемент критично важливий для забезпечення конфіденційності та захисту даних від несанкціонованого доступу. Шифрування має відбуватися як на етапі передачі даних, так і при їх зберіганні на серверах. Крім того, застосування сучасних методів аутентифікації та авторизації є необхідним для забезпечення того, що доступ до даних мають лише уповноважені користувачі.

Мережева інфраструктура відіграє ключову роль у забезпеченні стабільного та швидкого з'єднання між клієнтами та серверами. Мережева інфраструктура – це фізичні та програмні ресурси, необхідні для підтримки з'єднань між клієнтськими програмами та серверами даних [6]. Включає в себе мережеве обладнання (маршрутизатори, комутатори), телекомунікаційні канали (включаючи інтернет) та протоколи передачі даних, які забезпечують швидкий та надійний обмін даними і гарантує, що дані можуть передаватися ефективно та без затримок.

Нарешті, інтерфейси програмування додатків надають розробникам засоби для інтеграції функціоналу хмарних сховищ у власні додатки та послуги. Інтерфейси програмування додатків або API – це набір визначень та протоколів для створення та інтеграції програмного забезпечення. API в хмарних сховищах дозволяють розробникам підключати свої додатки до хмарних сервісів, використовувати їх функції для зберігання та обробки даних, та розробляти налаштовані рішення. Це дозволяє створювати налаштовані рішення, які можуть використовувати переваги хмарного зберігання, такі як гнучкість, масштабованість та доступність.

Комбінуючи ці елементи, хмарні сховища надають потужні та гнучкі рішення для зберігання та обробки великих обсягів даних, що є ключовим для сучасних цифрових екосистем.

1.2 Методи захисту даних у хмарних сервісах

1.2.1 Загальний огляд методів захисту даних

Забезпечення безпеки даних є важливою частиною управління хмарними сервісами. У сучасному цифровому світі, де обсяги даних невпинно зростають, а хмарні технології стають все більш популярними, захист інформації стає ключовим викликом. Несанкціонований доступ, втрати даних та зловмисні атаки є загрозами, з якими регулярно стикаються користувачі хмарних сервісів. Це ставить під загрозу не тільки конфіденційність, але й цілісність та доступність даних [7]. В цьому контексті розробка ефективних стратегій шифрування та впровадження надійних механізмів безпеки є критично важливими. Ключовими аспектами захисту даних у хмарних сервісах є:

– шифрування даних. Це основний інструмент захисту інформації в хмарі. Шифрування перетворює чутливі дані в зашифрований формат, який може бути розшифрований лише з використанням спеціального ключа. Шифрування захищає дані під час їх зберігання на хмарних серверах, а також під час передачі через Інтернет [8];

– управління доступом і аутентифікація. Обмеження доступу до даних та систем ідентифікації користувачів є критично важливими для запобігання несанкціонованого доступу. Методи, як-от багатофакторна аутентифікація та ролеве управління доступом, забезпечують, що лише уповноважені особи мають доступ до чутливої інформації;

– регулярне оновлення безпеки. Технології швидко змінюються, а з ними і загрози безпеці. Регулярне оновлення систем і впровадження оновлень безпеки допомагають захистити хмарні сервіси від відомих вразливостей;

– резервне копіювання та відновлення даних. Важливим елементом стратегії безпеки є створення надійних планів резервного копіювання та відновлення. Це гарантує, що дані можуть бути відновлені у випадку їх втрати або пошкодження;

– моніторинг та аудит безпеки. Постійний моніторинг і аналіз активності в хмарних сервісах допомагає виявляти підозрілу поведінку та вживати заходів для запобігання потенційним загрозам.

Ці заходи, взаємодіючи разом, формують багаторівневу систему захисту, яка допомагає мінімізувати ризики та забезпечує надійне та безпечне зберігання та обробку даних у хмарному середовищі.

1.2.2 Шифрування даних

Шифрування даних у технічному контексті включає перетворення читабельних даних у зашифрований формат за допомогою криптографічних алгоритмів. Шифрування – це процес перетворення даних в зашифрований формат, який неможливо прочитати без відповідного ключа шифрування [5]. Цей процес захищає конфіденційність даних під час їх зберігання на хмарних серверах та передачі через інтернет. Цей процес перетворює зрозумілу інформацію, таку як текстові документи або фінансові дані, у послідовності, що виглядають як випадкові символи та числа. Ця трансформація забезпечує безпеку даних, оскільки зробити їх знову читабельними можливо лише з використанням відповідного ключа шифрування.

Технічно, шифрування базується на математичних принципах та алгоритмах. Існують різні методи шифрування, такі як симетричне та асиметричне шифрування. У симетричному шифруванні використовується один і той же ключ для шифрування та розшифрування даних, тоді як асиметричне шифрування використовує пару ключів – публічний для шифрування та приватний для розшифрування.

Шифрування є особливо важливим у хмарних сервісах, оскільки забезпечує безпеку даних під час їх трансмісії через інтернет та при зберіганні на віддалених

серверах. Це захищає інформацію від несанкціонованого доступу, втручання або крадіжки, забезпечуючи конфіденційність та цілісність даних.

Розглянемо ключові аспекти шифрування, які використовуються в хмарних сервісах:

– види шифрування. Існують різні види шифрування, такі як шифрування на стороні клієнта та шифрування на стороні сервера. Шифрування на стороні клієнта означає, що дані шифруються до того, як вони відправляються на сервер, тоді як шифрування на стороні сервера відбувається після того, як дані вже знаходяться на хмарному сервері;

– управління ключами шифрування. Ефективне управління ключами є важливим аспектом шифрування. Це включає зберігання, видачу та оновлення ключів шифрування, а також забезпечення їх безпеки;

– протоколи шифрування. Використання стандартних та перевірених протоколів шифрування допомагає забезпечити надійний захист даних. Важливо вибирати протоколи, які відомі своєю міцністю та надійністю;

– виклики та обмеження шифрування. Хоча шифрування є ефективним засобом захисту, існують виклики, пов'язані з управлінням ключами та можливістю доступу до зашифрованих даних для законних цілей.

Розглянемо детальніше методи шифрування, що використовуються у хмарних сервісах, кожен з яких має свої специфічні особливості та області застосування. Це допоможе нам краще зрозуміти, як різні технології шифрування впливають на безпеку даних, що зберігаються та обробляються у хмарі. Ось декілька з найбільш використовуваних методів:

– AES (Advanced Encryption Standard) є широко використовуваним стандартом шифрування, який застосовує симетричний ключ для шифрування та розшифрування даних. Цей метод включає перетворення даних у блоки по 128 біт кожен, розширення ключа, додавання раундового ключа, заміну байтів, зсув рядків, змішування стовпців та додавання ключа з етапу розширення ключа до блокового шифру з останнього кроку. AES забезпечує високий рівень безпеки і часто використовується для захисту конфіденційної інформації;

– RSA (Rivest–Shamir–Adleman) – це система асиметричного шифрування, яка використовує пару ключів: публічний для шифрування та приватний для розшифрування. Процес включає вибір двох простих чисел для генерації ключів, розрахунок добутку та тотієнта, вибір публічного та приватного ключа, а також шифрування та розшифрування повідомлень. RSA є однією з основ криптографічних систем, використовуваних у багатьох безпечних Інтернет-протоколах;

– TLS/SSL (Transport Layer Security/Secure Sockets Layer) забезпечують безпечну передачу даних між клієнтом та сервером в інтернеті, використовуючи комбінацію симетричного та асиметричного шифрування. Процес включає обмін SSL-сертифікатами, вимогами до шифрованих наборів та випадковими даними для створення сеансових ключів, а також встановлення криптографічних алгоритмів та сеансових ключів;

– Шифрування на основі ECC (Elliptic Curve Cryptography) забезпечує високий рівень безпеки з використанням менших ключів, ніж RSA, роблячи його ефективним у мобільних та інших обмежених середовищах;

– SHA-256 (Secure Hash Algorithm 256-bit) – це криптографічна хеш-функція, яка не використовується для розшифрування даних, але часто застосовується разом з іншими методами шифрування для перевірки цілісності даних. Процес включає доповнення вхідних даних до певної довжини, розділення на блоки, встановлення початкових хеш-значень, основний цикл хешування та отримання кінцевого хешу.

Шифрування в хмарних сервісах використовує криптографічні алгоритми для перетворення зрозумілих даних у зашифровані послідовності, доступні тільки з відповідним ключем. Симетричне шифрування, як AES, використовує один ключ для обох процесів, тоді як асиметричне, як RSA, має окремі ключі для шифрування та розшифрування. Також використовуються TLS/SSL для захищеного з'єднання між клієнтом і сервером та ECC для більш ефективного шифрування в обмежених середовищах.

Ці методи гарантують безпеку даних під час передачі та зберігання, захищаючи їх від несанкціонованого доступу та утручань. Вони відіграють ключову роль у забезпеченні безпеки інформації в хмарних обчисленнях, що є важливим у сучасному цифровому світі.

1.2.3 Аутентифікація та авторизація

Аутентифікація та авторизація відіграють фундаментальну роль у забезпеченні безпеки хмарних сервісів, встановлюючи ідентичність користувачів та визначаючи їх доступ до ресурсів. Аутентифікація - це процес перевірки ідентичності користувача або системи. Під час аутентифікації система переконується, що особа або система, яка намагається отримати доступ, дійсно є тією, за кого себе видає. Авторизація - це процес визначення прав та привілеїв користувача після його успішної аутентифікації в системі. Вона визначає, що конкретно користувач може робити в системі, наприклад, які дані він може переглядати, на які ресурси має доступ, які операції може виконувати [9].

Процес аутентифікації включає не тільки базові механізми, такі як імена користувачів та паролі, але й більш складні методи, такі як двофакторна аутентифікація, використання біометричних даних та одноразових паролів. Це гарантує, що доступ до системи надається тільки перевіреним особам. Аутентифікація в хмарних сервісах, рисунок 1.1, може також інтегрувати механізми, які враховують контекст користувача, такі як місцезнаходження, час входу та інші параметри, щоб забезпечити більш надійний рівень захисту.

Рис. 1.1 Процес аутентифікації

Авторизація, рисунок 1.2, у свою чергу, визначає рівень доступу, який надається аутентифікованому користувачеві. Це включає ролеву модель, де користувачам присвоюються певні ролі, кожна з яких має визначений набір прав та привілеїв. Крім того, авторизація може ґрунтуватися на політиках доступу, які деталізовано визначають можливості користувачів в системі. Важливим аспектом є гнучкість системи авторизації, яка дозволяє адаптуватися до змінних потреб та вимог безпеки в хмарних сервісах.

Рис. 1.2 Процес аутентифікації та авторизації

Ефективне управління аутентифікацією та авторизацією в хмарних сервісах вимагає ретельного підходу, з урахуванням як технологічних аспектів, так і політик безпеки. Це включає регулярні оновлення, аудит системи та впровадження заходів для захисту від внутрішніх та зовнішніх загроз. Забезпечення безпеки даних в хмарі - це постійний процес, який вимагає уваги до деталей, адаптивності до нових викликів та готовності швидко реагувати на зміни в кіберзагрозах [10].

Це важливий аспект безпеки в хмарних сервісах, який може використовувати різноманітні методи, включаючи двофакторну аутентифікацію та біометричні дані. Ефективне управління цими процесами вимагає технологічного підходу та політики безпеки, а також постійного оновлення та адаптації до нових загроз. Забезпечення безпеки даних в хмарних сервісах є неперервним процесом, який потребує уваги до деталей та готовності реагувати на зміни в кіберзагрозах.

1.3 Аналіз надійності та масштабованості хмарних систем

Хмарні системи сьогодні стоять на передовій цифрової трансформації, визначаючи нові виміри надійності та масштабованості в інформаційних технологіях. Аналізуючи їх вплив, важливо розуміти, як інновації в хмарних технологіях сприяли створенню більш гнучких та ефективних систем. Ці системи не лише підвищують потенціал зберігання даних і обчислювальних можливостей, але й відкривають шлях для безпрецедентного рівня масштабування та доступності сервісів. Розвиток від централізованих до розподілених архітектур і назад до централізованих, але вже в хмарі, відображає цю динаміку.

В контексті цієї швидкої еволюції, важливо зазначити, що кожна ера IT радикально переформатувала способи, якими організації обробляють дані, проектують додатки та розгортають інфраструктуру. Зокрема, хмарні обчислення ініціювали значні зміни в архітектурі підприємств та програмного забезпечення. Завдяки віртуалізації IT-інфраструктури, упаковці розробницьких інструментів,

сервісів та додатків, хмарні обчислення спричинили глибоке переосмислення дизайну додатків.

Рис. 1.3 Еволюція інфраструктури додатків

Рисунок 1.3 демонструє, як змінювалася інфраструктура додатків з часом, від 1960-х до 2020-х років. Починаючи з ери мейнфреймів у 1960-х, які символізують централізовану модель обчислень, де великі, потужні комп'ютери використовувалися для виконання усіх обчислювальних завдань. У 1970-х мінікомп'ютери запропонували більш доступну альтернативу, що дозволила меншим організаціям інтегрувати технології в свої операції.

У 1980-х робочі станції принесли обчислювальну потужність безпосередньо до робочих місць фахівців, дозволяючи більш комплексні технічні та дизайнерські завдання. Наступне десятиліття – 1990-ті – відзначилося появою персональних комп'ютерів і серверів, що знаменувало перехід до більш децентралізованої моделі обчислень і сприяло розвитку мереж Інтернет.

2000-ті внесли нову еру з використанням віртуальних машин і серверів, які забезпечували кращу масштабованість та ефективність IT-інфраструктури. І, нарешті, 2010-ті роки стали свідками впровадження хмарних рішень як сервісу (IaaS), що дозволило організаціям орендувати інфраструктуру від провайдерів послуг, замість купівлі та утримання власного обладнання.

2020-ті роки відкрили нові горизонти з платформи, як сервісу (PaaS), контейнеризацією, мікросервісами та безсерверними архітектурами, що подальше спростили розробку, розгортання та масштабування додатків. Ці технології

підтримують швидку інновацію та адаптацію, дозволяючи організаціям швидко реагувати на змінні ринкові вимоги та потреби користувачів [2].

Сучасні хмарні системи демонструють новітні стандарти надійності, впроваджуючи розподілені архітектури, які забезпечують високу доступність та стійкість до збоїв. Автоматизовані процеси відновлення після збоїв та регулярне резервне копіювання даних знижують ризики втрати інформації. Хмарні провайдери, як Amazon Web Services чи Google Cloud, використовують географічно розподілені центри даних, що дозволяє клієнтам мати безперервний доступ до своїх додатків та даних, навіть у випадку локальних катастроф.

Масштабованість хмарних систем є їх вирішальною перевагою. Масштабованість – це здатність хмарної інфраструктури ефективно адаптуватися до зростаючих або зменшених вимог до обчислювальних ресурсів без втрати продуктивності або якості обслуговування. Система може обробляти збільшення чи зменшення навантаження, автоматично додаючи або віднімаючи ресурси, такі як обчислювальна потужність, пам'ять і сховище даних. Динамічне масштабування, яке відбувається автоматично відповідно до потреб користувачів, забезпечує оптимальне використання ресурсів, дозволяючи підприємствам платити за фактично використане, а не за потенційно необхідне. Це не тільки зменшує витрати, але й дозволяє компаніям швидко адаптуватися до змін на ринку, збільшуючи або зменшуючи ресурси в межах лічених хвилин.

Трансформація від традиційних централізованих систем до розподілених хмарних сервісів також підвищила важливість віртуалізації. Віртуалізація дозволяє створювати і управляти віртуальними машинами, що імітують фізичні сервери, але з більшою гнучкістю та ефективністю. Це сприяло появі контейнерів і мікросервісів, які вдосконалюють процес розробки та розгортання додатків, дозволяючи індивідуальні компоненти незалежно один від одного масштабувати та оновлювати.

Без серверні обчислення, які з'явилися в останнє десятиліття, подальше усувають необхідність управління інфраструктурою, дозволяючи розробникам фокусуватися на коді. Ця модель пропонує ідеальну масштабованість, оскільки

сервери автоматично розгортаються та масштабуються хмарними провайдерами в залежності від потреби в обчислювальних ресурсах.

У цілому, еволюція хмарних систем підкреслює швидкий розвиток технологій та необхідність неперервної інновації для підтримки глобальної конкурентоспроможності. Це створює нові можливості для бізнесів різного розміру, дозволяючи їм бути більш гнучкими та адаптивними до змін у своїх галузях.

2 ДОСЛІДЖЕННЯ СУЧАСНИХ ПІДХОДІВ ТА ІНСТРУМЕНТІВ У ХМАРНИХ СХОВИЩАХ

2.1 Порівняльний аналіз хмарних технологій

Розвиток хмарних технологій став віхою у сфері цифрових інновацій, відкриваючи нові можливості для зберігання, обробки та аналізу даних. Ці технології забезпечують компаніям та індивідуальним користувачам доступ до потужних обчислювальних ресурсів через інтернет, дозволяючи їм використовувати розширені можливості хмарних сервісів без значних капіталовкладень у власну інфраструктуру. Зосереджуючись на архітектурі хмарних сховищ, цей розділ проводить порівняльний аналіз ключових провайдерів цих сервісів, висвітлюючи їх основні особливості, стратегії та потенціал для різних застосувань.

Основою хмарних сховищ є кілька фундаментальних принципів, таких як масштабованість, еластичність та безпека. Ці характеристики забезпечують не тільки ефективне управління даними, але й гарантують високий рівень доступності та надійності сервісу. Розуміння цих аспектів є ключовим для вибору найбільш підходящого хмарного рішення, що відповідає специфічним потребам користувачів.

Розглянемо п'ять провідних хмарних платформ - AWS (Amazon Web Services), Azure (Microsoft), Google Cloud Platform, IBM Cloud та Oracle Cloud. Кожна з цих платформ має свої унікальні особливості, що відображаються в їхній архітектурі, підходах до безпеки, масштабування та загальної ефективності.

AWS Architecture відрізняється своєю надійністю та гнучкістю, пропонуючи різноманітні послуги від базового хостингу до розширених рішень у сфері штучного інтелекту та машинного навчання. Azure Architecture, з іншого боку, фокусується на безшовній інтеграції з іншими продуктами Microsoft, що робить її

ідеальною для корпоративних клієнтів, які вже використовують інші продукти цієї компанії.

Google Cloud Platform вирізняється інноваціями в області обробки великих даних та штучного інтелекту, надаючи потужні інструменти для аналізу та обробки інформації. IBM Cloud зосереджений на наданні корпоративних рішень, зокрема, у сферах, де потрібна висока ступінь спеціалізації та індивідуалізації. Oracle Cloud, у свою чергу, пропонує інтегровані рішення, що комбінують хмарні технології з традиційними базами даних.

Зараз детально проаналізуємо особливості кожної платформи, аналізуючи їх сильні та слабкі сторони, а також виявляючи основні фактори, які роблять їх привабливими для певних типів користувачів та бізнес-сценаріїв. Це допоможе отримати глибоке розуміння того, як кожна з цих платформ може бути використана для задоволення конкретних потреб в хмарних рішеннях.

Розвиток хмарних технологій змінив підхід до управління даними та комп'ютерної інфраструктури, і в цьому контексті Amazon Web Services (AWS) займає провідне місце завдяки своїй інноваційній архітектурі. Архітектура AWS є втіленням гнучкості, масштабованості та безпеки, що необхідні для ефективної хмарної платформи.

Основою AWS є її глобальна інфраструктура, яка включає множину дата-центрів, розташованих у різних географічних регіонах і зонах доступності. Ця розподілена структура гарантує високий рівень довговічності та доступності даних, а також знижує затримки, забезпечуючи швидкий доступ до даних незалежно від географічного розташування користувачів.

У сфері обчислювальних послуг AWS пропонує Amazon EC2 (Elastic Compute Cloud), що дозволяє користувачам ефективно віртуалізувати обчислювальні ресурси. Вибір інстансів EC2 може бути оптимізований під конкретні потреби, від базових до високопродуктивних обчислень, надаючи користувачам необхідну гнучкість.

Для зберігання даних AWS надає широкий спектр послуг, як-от Amazon S3 для об'єктного сховища, що забезпечує високу надійність та доступність даних,

Amazon EBS для блокових сховищ, які використовуються разом з EC2, та Amazon Glacier для довгострокового зберігання великих обсягів даних.

Мережеві послуги в AWS, як Amazon VPC, дозволяють створювати ізольовану мережу в межах AWS, надаючи повний контроль над IP-адресами, підмережами, маршрутизацією та доступом до Інтернету. Це забезпечує додатковий рівень безпеки та гнучкості для корпоративних клієнтів.

Безпека в AWS забезпечується через комплексні інструменти, такі як AWS Identity and Access Management (IAM), які дозволяють контролювати доступ до ресурсів AWS. Додаткові послуги, такі як Amazon GuardDuty, Amazon Inspector та AWS Shield, посилюють захист від зовнішніх загроз.

Інтеграція та управління в AWS підкріплюється набором інструментів, що дозволяють керувати інфраструктурою як кодом, стежити за ресурсами та автоматично масштабувати їх в залежності від потреб. AWS CloudFormation, AWS CloudWatch та AWS Auto Scaling є ключовими інструментами в цьому аспекті [16].

Архітектура AWS, рисунок 2.1, підходить для широкого спектра застосувань – від веб-сайтів та мобільних додатків до складних корпоративних систем і машинного навчання. Гнучкість та широкий спектр послуг роблять AWS вибором для компаній, що шукають надійне та масштабоване хмарне рішення.

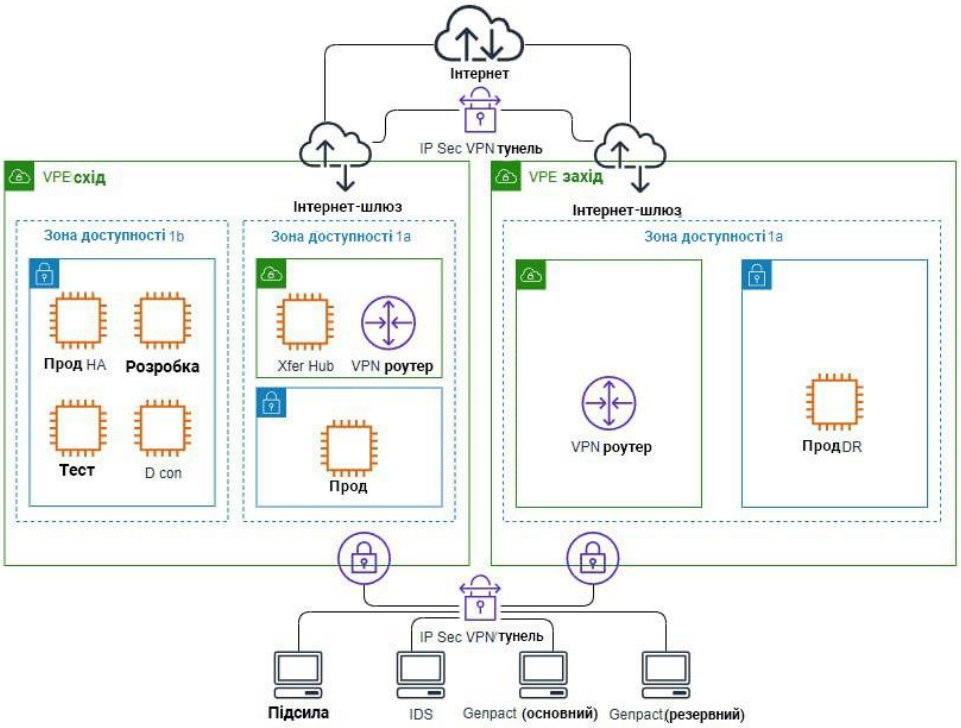


Рис. 2.1 Хмарна архітектура Amazon Web Services

Google Cloud Platform (GCP) відома своєю інноваційною архітектурою та передовими технологічними рішеннями, що робить її однією з провідних платформ у світі хмарних обчислень. GCP надає різноманітні послуги, забезпечуючи гнучкість, масштабованість і високий рівень безпеки для своїх користувачів.

Однією з ключових особливостей GCP є її глобальна інфраструктура, рисунок 2.2, яка включає численні центри обробки даних, розташовані по всьому світу. Ця розподілена мережа гарантує низьку затримку та високу доступність послуг, що особливо важливо для бізнесів з глобальними потребами. Інфраструктура GCP розділена на регіони та зони доступності, що дозволяє користувачам оптимізувати використання ресурсів з урахуванням географічного розташування [17].

У сфері обчислювальних послуг GCP пропонує Google Compute Engine, що надає можливість віртуалізації обчислювальних ресурсів. Це дає користувачам

велику гнучкість у виборі конфігурацій віртуальних машин, відповідно до їхніх потреб та вимог. Для зберігання даних GCP пропонує різні рішення, включаючи Google Cloud Storage для об'єктного сховища, Cloud SQL для управління базами даних SQL і Google Cloud Bigtable для великомасштабної обробки даних в реальному часі. Ці послуги забезпечують надійність, високу продуктивність та гнучкість у зберіганні та обробці даних.

Важливим аспектом GCP є також мережеві послуги, такі як Google Cloud VPC, що дозволяє створювати ізольовану мережу в межах облачної інфраструктури. Це надає додатковий рівень контролю та безпеки, дозволяючи користувачам точно налаштовувати мережеві налаштування згідно зі своїми потребами. Безпека в GCP забезпечується за допомогою різноманітних інструментів, таких як Google Cloud Identity & Access Management для контролю доступу до ресурсів, а також інших сервісів для управління безпекою інфраструктури.

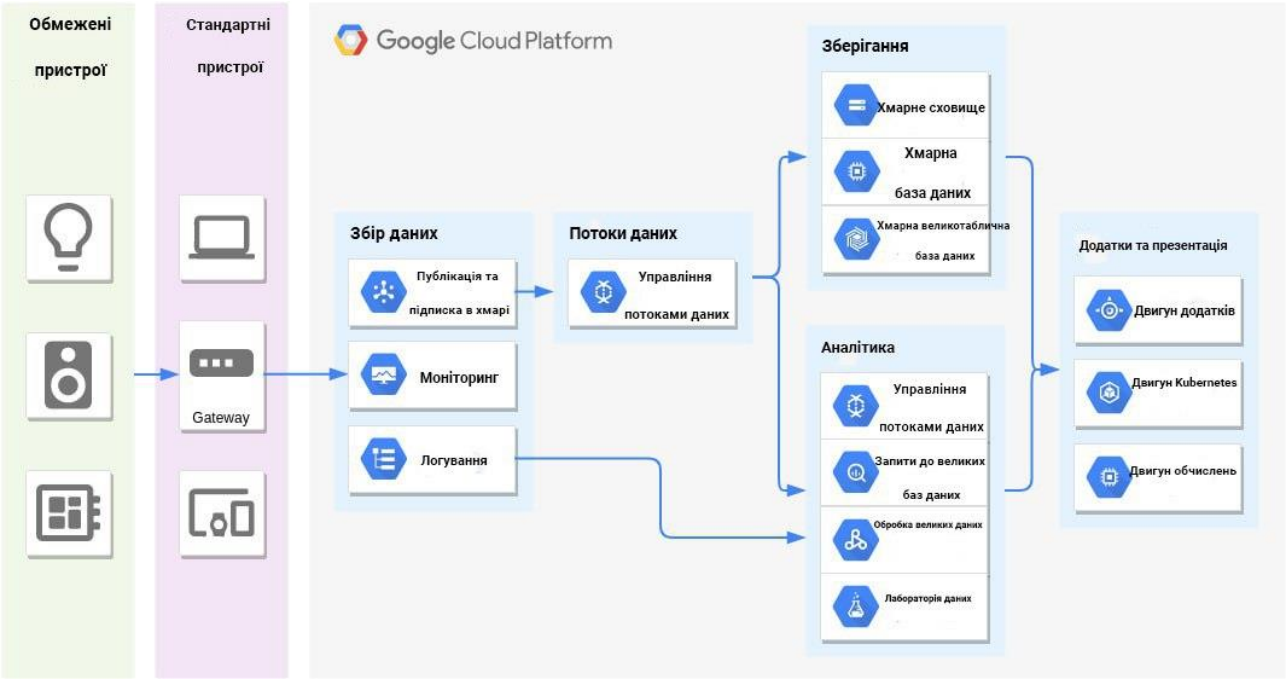


Рис. 2.2 Хмарна архітектура Google Cloud Platform

Інструменти для інтеграції та управління в GCP, такі як Google Cloud Deployment Manager для управління інфраструктурою як кодом, Google

Stackdriver для моніторингу та діагностики, та Google Kubernetes Engine для управління контейнерами, роблять GCP ефективним рішенням для комплексного управління хмарними ресурсами.

GCP ідеально підходить для різноманітних застосувань, від веб-хостингу до складних проектів у сфері аналізу великих даних та машинного навчання. Її гнучкість та інноваційні технологічні рішення роблять GCP вибором для бізнесів, які прагнуть використовувати передові обчислювальні можливості.

Microsoft Azure, один із провідних гравців у світі хмарних обчислень, вирізняється своєю комплексною архітектурою та глибокою інтеграцією з іншими продуктами Microsoft. Azure пропонує широкий спектр обчислювальних, зберігаючих та мережевих послуг, що робить його ідеальним рішенням для бізнесів будь-якого розміру. Однією з відмітних особливостей Azure є його глобальна інфраструктура, що складається з численних центрів обробки даних, розташованих у різних регіонах світу. Ця інфраструктура, рисунок 2.3, забезпечує високу доступність та довговічність даних, а також дозволяє клієнтам ефективно розподіляти ресурси [18].

У сфері обчислювальних послуг Azure пропонує Virtual Machines для віртуалізації обчислень та Azure Kubernetes Service для оркестрації контейнерів, забезпечуючи гнучкість та масштабованість обчислювальних ресурсів. Azure також включає інноваційні рішення, такі як Azure Functions, які дозволяють користувачам запускати код без потреби управління серверними ресурсами. Для зберігання даних Azure надає різноманітні опції, включаючи Azure Blob Storage для об'єктного зберігання, Azure SQL Database для управління реляційними базами даних, та Azure File Storage для забезпечення файлового зберігання у хмарі. Ці послуги забезпечують надійність, високу продуктивність та гнучкість для зберігання та обробки даних.

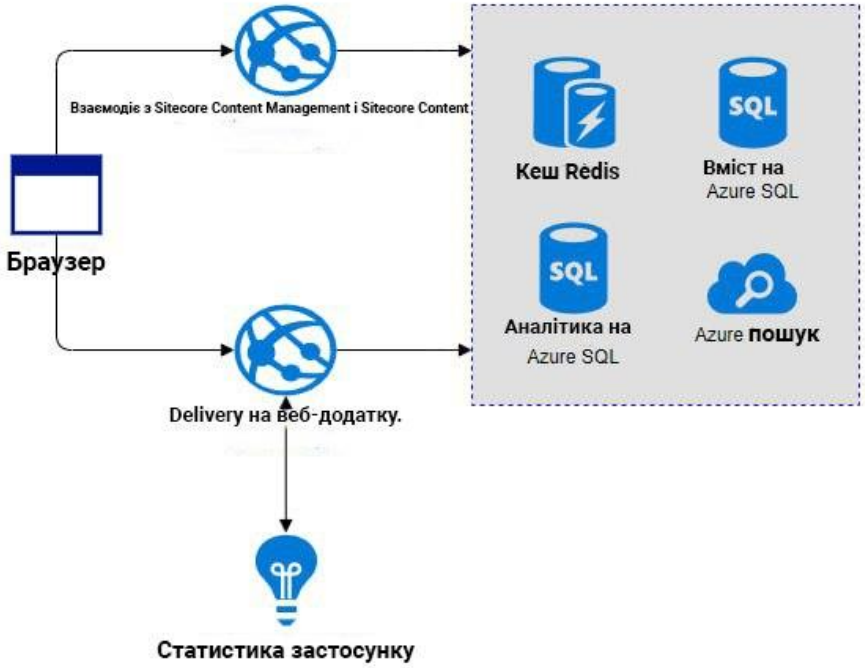


Рис. 2.3 Хмарна архітектура Microsoft Azure

В області мережевих сервісів Azure пропонує Azure Virtual Network, що дозволяє створювати повністю ізольовані та налаштовані мережеві рішення у хмарі. Це надає додаткову гнучкість та безпеку для корпоративних клієнтів. З точки зору безпеки Azure розроблений із врахуванням вимог безпеки та конфіденційності. За допомогою Azure Active Directory та різних інструментів безпеки, таких як Azure Sentinel та Azure Security Center, користувачі мають можливість контролювати доступ та виявляти загрози в реальному часі.

Інтеграція та управління в Azure спрощені завдяки таким інструментам, як Azure Resource Manager для управління ресурсами і Azure Monitor для моніторингу стану ресурсів та додатків. Ці інструменти дозволяють користувачам легко керувати своїми хмарними ресурсами. Azure ідеально підходить для широкого спектра бізнес-застосувань, від веб-хостингу до розробки складних корпоративних додатків, завдяки своїй інтеграції з іншими продуктами Microsoft і глибокої підтримки для розробників.

IBM Cloud вирізняється у світі хмарних обчислень завдяки своїй сильній орієнтації на корпоративні потреби та інноваційні технології. Ця платформа надає широкий спектр послуг, що охоплюють обчислювальні можливості, зберігання даних, аналітику та штучний інтелект, забезпечуючи комплексні рішення для бізнесу. Основою IBM Cloud є її глобальна інфраструктура, яка включає мережу центрів обробки даних, розташованих по всьому світу. Це забезпечує високу доступність та надійність послуг, а також дозволяє ефективно масштабувати ресурси відповідно до потреб клієнтів.

У сфері обчислювальних послуг IBM Cloud пропонує IBM Cloud Virtual Servers, які надають гнучкість та масштабованість віртуалізованих обчислювальних ресурсів, рисунок 2.4. Крім того, IBM Cloud Kubernetes Service дозволяє ефективно управляти контейнеризованими застосуваннями, що є важливим для сучасних обчислювальних вимог.

Що стосується зберігання даних, IBM Cloud пропонує різні варіанти, включаючи IBM Cloud Object Storage для об'єктного зберігання великих обсягів даних, IBM Cloud Block Storage для високопродуктивного блокового зберігання та IBM Cloud File Storage для гнучкого файлового зберігання. Ці рішення забезпечують ефективне управління даними та оптимізацію зберігання.

У площині мережевих послуг IBM Cloud надає IBM Cloud Virtual Private Cloud, що дозволяє створювати безпечні та ізольовані мережеві середовища в хмарі. Такі рішення надають додаткову гнучкість та контроль над мережевими ресурсами. IBM Cloud підкріплює безпеку своїх сервісів за допомогою комплексних інструментів та протоколів. IBM Cloud Identity and Access Management забезпечує контроль доступу до ресурсів, а послуги безпеки, такі як IBM Cloud Security Advisor, допомагають виявляти та управляти загрозами.

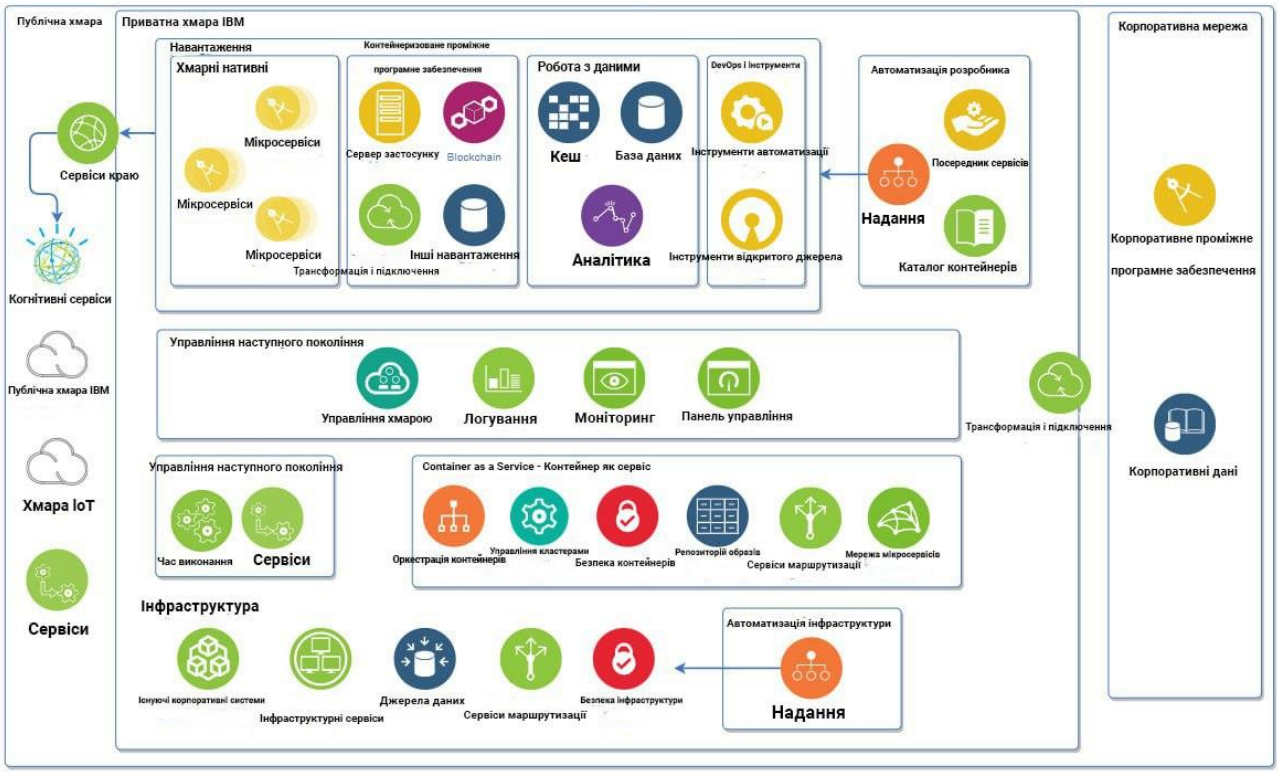


Рис. 2.4 Хмарна архітектура IBM Cloud

Інструменти для інтеграції та управління в IBM Cloud, такі як IBM Cloud Resource Manager та IBM Cloud Monitoring, спрощують управління хмарними ресурсами та моніторинг стану системи, що дозволяє користувачам ефективно керувати своїм хмарним середовищем. IBM Cloud особливо сильна в сегментах, де потрібні спеціалізовані рішення, такі як штучний інтелект та аналітика даних, завдяки інтеграції з такими продуктами як IBM Watson. Це робить платформу ідеальною для компаній, які шукають інноваційні та інтегровані хмарні рішення [19].

Oracle Cloud відіграє ключову роль у світі хмарних обчислень, особливо завдяки своїй зосередженості на інтеграції хмарних послуг з традиційними даними та додатками. Ця платформа пропонує унікальну комбінацію обчислювальних ресурсів, зберігання даних, мережевих можливостей та спеціалізованих рішень для обробки великих даних і штучного інтелекту.

Основу Oracle Cloud складає її глобальна інфраструктура, рисунок 2.5, яка включає декілька дата-центрів, розташованих у різних частинах світу. Ця розподілена мережа забезпечує високий рівень доступності та надійності послуг,

що дозволяє забезпечити ефективне масштабування ресурсів та зниження затримок. В області обчислювальних послуг Oracle Cloud пропонує Oracle Cloud Infrastructure (OCI) Virtual Machines, які надають потужні та гнучкі віртуалізовані обчислювальні ресурси. OCI також включає контейнерні сервіси та функції, такі як Oracle Kubernetes Service, що дозволяє ефективно управляти контейнеризованими застосуваннями [20].

У сфері зберігання даних Oracle Cloud надає широкий спектр опцій, включаючи Oracle Cloud Storage для об'єктного зберігання, Block Storage для високопродуктивного блокового зберігання, та File Storage для гнучкого файлового зберігання. Ці рішення оптимізують управління даними та забезпечують високу продуктивність. Мережеві послуги в Oracle Cloud, включаючи Oracle Cloud Virtual Cloud Network (VCN), дозволяють створювати повністю ізольовану та налаштовану мережеву інфраструктуру, забезпечуючи додатковий рівень безпеки та контролю для корпоративних клієнтів.

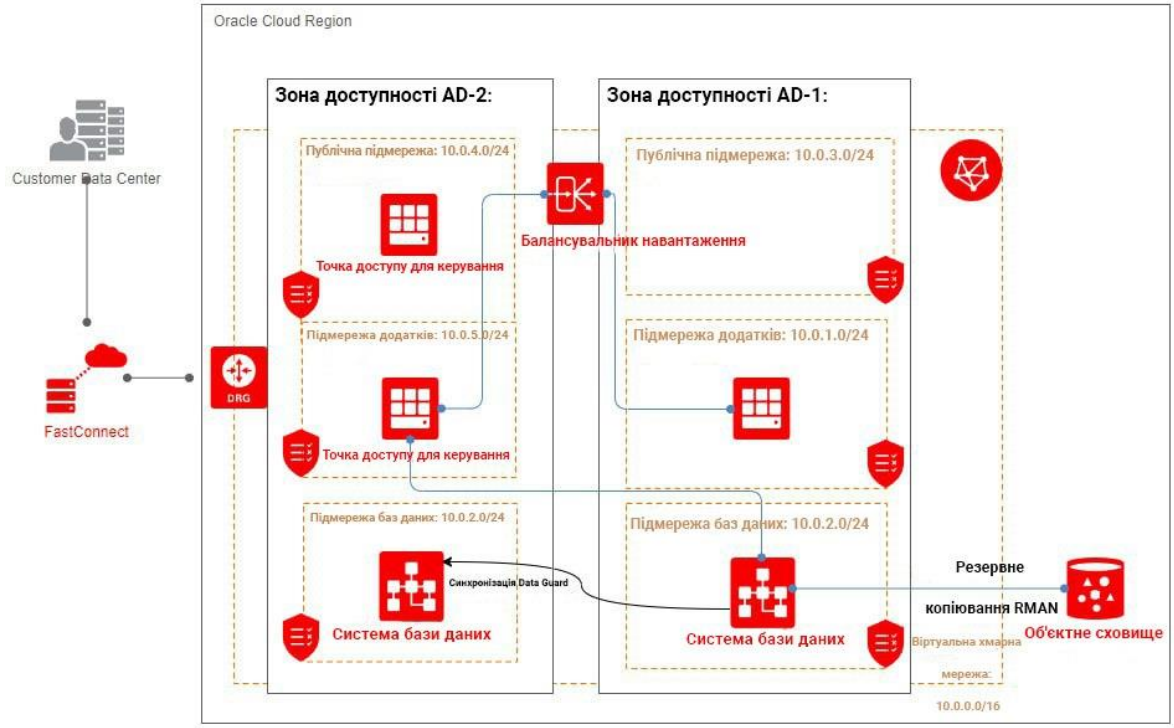


Рис. 2.5 Хмарна архітектура Oracle Cloud

Особлива увага в Oracle Cloud приділяється безпеці, що включає застосування сучасних інструментів та протоколів для контролю доступу та моніторингу безпеки. Oracle Cloud Identity and Access Management дозволяє ретельно контролювати доступ до ресурсів, а додаткові інструменти безпеки, такі як Oracle Cloud Security Monitoring and Analytics, надають розширені можливості для захисту інфраструктури.

Що стосується інтеграції та управління, Oracle Cloud пропонує інструменти, такі як Oracle Cloud Resource Manager для управління інфраструктурою як кодом та Oracle Cloud Monitoring для стеження за станом ресурсів, спрощуючи управління хмарними сервісами. Oracle Cloud є особливо привабливим рішенням для компаній, що шукають інтегровані хмарні послуги, які легко поєднуються з їхніми існуючими базами даних і додатками. Його спеціалізовані рішення для обробки великих даних, штучного інтелекту та автоматизованого машинного навчання роблять його вибором для бізнесів, які прагнуть до інновацій.

Після аналізу провідних хмарних платформ, таких як AWS, Google Cloud Platform, Microsoft Azure, IBM Cloud та Oracle Cloud, можна зробити деякі ключові висновки. AWS вирізняється своєю масштабованістю та надійністю, охоплюючи широкий спектр бізнес-потреб. Google Cloud Platform фокусується на інноваціях у сфері великих даних та штучного інтелекту, ідеально підходячи для технологічних розробок. Microsoft Azure пропонує глибоку інтеграцію з продуктами Microsoft, роблячи його вибором для користувачів цієї екосистеми. IBM Cloud спеціалізується на корпоративних рішеннях, особливо в області AI та аналітики, тоді як Oracle Cloud ідеально підходить для інтеграції з традиційними базами даних і додатками. Вибір між цими платформами залежить від конкретних потреб бізнесу та існуючої інфраструктури.

2.2 Безпека даних у хмарних технологіях: роль регуляторів та відповідальність провайдерів

З огляду на швидкий розвиток технологій у сфері хмарних обчислень, питання безпеки даних набуває особливої актуальності. Оскільки хмарні платформи, такі як AWS, Google Cloud Platform, Microsoft Azure, IBM Cloud та Oracle Cloud, відіграють ключову роль у зберіганні та обробці величезних обсягів даних, вони також впроваджують передові підходи та технології для забезпечення їх безпеки. Ці платформи застосовують ряд інноваційних стратегій, щоб відповідати постійно зростаючим вимогам до конфіденційності та захисту інформації у хмарному середовищі.

Політики безпеки та регуляторні вимоги також грають значну роль у хмарних платформах. Компанії, які використовують хмарні сервіси, повинні впевнитися, що їх провайдери дотримуються відповідних законодавчих та галузевих стандартів. Регулятори – це організації або установи, які відповідають за нагляд, контроль та впровадження законодавчих норм у різних сферах, включаючи захист даних та приватності. Вони встановлюють стандарти та правила, забезпечують їх дотримання та накладають штрафи за порушення.

Регулятори в сфері захисту даних та приватності відіграють вирішальну роль у сучасному цифровому світі, особливо з урахуванням стрімкого розвитку хмарних технологій та збільшення обсягів оброблюваних даних. Одними з ключових регулятивів у цій сфері є GDPR (Загальний регламент про захист даних) та HIPAA (Закон про переносимість та відповідальність страхування здоров’я).

GDPR, прийнятий Європейським Союзом, встановлює норми, які регулюють обробку особистих даних усередині ЄС та передачу цих даних за його межі. Цей регламент визначає особисті дані як будь-яку інформацію, що може бути використана для ідентифікації фізичної особи, включаючи ім'я, адресу, дані електронної пошти, а також більш чутливі дані, такі як біометричні дані. Основними принципами GDPR є прозорість у зборі та обробці даних, обмеження

мети, точність даних, обмеження зберігання, цілісність, конфіденційність та підзвітність. Компанії, що обробляють дані громадян ЄС, повинні забезпечити високий рівень захисту цих даних, а також гарантувати права осіб на доступ, виправлення, видалення та переносимість їхніх даних.

З іншого боку, HIPAA – американський закон, який регулює захист медичної інформації. Цей закон став особливо актуальним з огляду на необхідність захисту чутливої медичної інформації, що обробляється лікарнями, страховими компаніями, та іншими медичними установами. HIPAA вимагає від цих організацій впровадження адекватних адміністративних, фізичних та технічних заходів для захисту конфіденційності та безпеки медичної інформації. Він також регулює обмін медичною інформацією між різними сторонами, вимагаючи від них дотримання певних стандартів конфіденційності та безпеки.

Невиконання вимог GDPR та HIPAA може призвести до серйозних юридичних наслідків, включаючи значні штрафи. Наприклад, GDPR передбачає штрафи до 20 мільйонів євро або до 4% від загального річного обороту компанії, залежно від того, яка сума більша, за серйозні порушення. HIPAA також передбачає штрафи за недотримання, що можуть сягати сотень тисяч доларів за кожне порушення. Важливо розуміти, що ці регулятиви мають глобальний вплив. Наприклад, компанії, розташовані поза ЄС, але які обробляють дані громадян ЄС, також підпадають під дію GDPR. Таким чином, міжнародні компанії, включаючи провайдерів хмарних послуг, повинні бути особливо уважними до дотримання цих регуляцій.

Хмарні платформи, такі як AWS, Google Cloud, Azure, IBM Cloud, та Oracle Cloud, враховують ці вимоги при розробці своїх політик безпеки та заходів захисту даних. Вони впроваджують різноманітні технологічні та організаційні механізми для забезпечення відповідності своїх послуг вимогам GDPR, HIPAA та інших регуляторних стандартів. Розглядаючи кожну з них окремо, можна побачити, що кожна з них має свої унікальні підходи до захисту даних. AWS зосереджується на масштабованих і гнучких рішеннях безпеки, що дозволяє їм ефективно реагувати на різноманітні виклики безпеки. Google Cloud Platform

використовує свої передові технології для аналітики безпеки, що допомагає їм у виявленні та нейтралізації потенційних загроз. Microsoft Azure вирізняється своєю інтеграцією з іншими продуктами Microsoft, надаючи додаткові можливості для безпеки, особливо у взаємодії з корпоративними середовищами. IBM Cloud фокусується на корпоративних рішеннях та використанні штучного інтелекту для підвищення рівня захисту даних. Нарешті, Oracle Cloud пропонує сильну інтеграцію з традиційними базами даних, що забезпечує додатковий рівень безпеки.

В контексті швидкого розвитку хмарних обчислень, безпека сховищ даних, що пропонуються компаніями на кшталт AWS, Google Cloud, Azure, IBM Cloud та Oracle Cloud, стає ключовим пріоритетом. Ці хмарні сховища відіграють не тільки роль зберігачів великих масивів даних, але й стають гарантами їх безпеки та цілісності. Важливість дотримання міжнародних стандартів, таких як GDPR та HIPAA, в цьому контексті не можна недооцінювати, адже вони закладають основу для надійного захисту інформації [25].

Роль регуляторів у цьому процесі є вирішальною, оскільки вони встановлюють межі та критерії, яким повинні відповідати хмарні сховища. Порушення цих норм може призвести до значних фінансових втрат та підриву довіри користувачів. Кожна з названих хмарних платформ реалізує свій унікальний підхід до забезпечення безпеки даних. Вони використовують передові технології та інноваційні методи для гарантування захисту інформації, що зберігається на їхніх серверах. Від масштабованих та гнучких рішень безпеки в AWS до сучасних аналітичних засобів в Google Cloud та комплексного підходу в Azure, кожна платформа вносить свій вклад у створення безпечного хмарного середовища.

Таким чином, хмарні сховища даних перетворюються на фортеці інформаційної безпеки, де сучасні технології та строгі регуляторні стандарти злиті разом для забезпечення надійного захисту даних. Це підкреслює не тільки технічну готовність, але й відповідальний підхід хмарних провайдерів до захисту цінної інформації своїх користувачів.

2.3 Технологічні аспекти сучасних інструментів безпеки в хмарних сховищах

Важливість питання безпеки посилюється з огляду на широке використання хмарних платформ для зберігання та обробки великих обсягів даних у різних секторах бізнесу та урядових установах. Основні хмарні платформи, такі як AWS, Google Cloud, Azure, IBM Cloud та Oracle Cloud, розробляють та впроваджують передові стратегії та технології безпеки для забезпечення захисту інформації, що зберігається в хмарі. Ці стратегії включають в себе використання шифрування даних, реалізацію багаторівневих систем автентифікації та моніторингу безпеки, а також адаптацію до зростаючих вимог щодо конфіденційності та захисту інформації.

Хмарні провайдери також активно впроваджують вимоги та стандарти, що встановлюються міжнародними та національними регуляторними органами. Ці норми встановлюють суворі вимоги до обробки та зберігання особистих та медичних даних, вимагаючи від хмарних платформ забезпечити високий рівень захисту цих даних. Важливо, що недотримання цих регуляторних стандартів може призвести до значних фінансових штрафів та юридичних наслідків[16].

У контексті розвитку сучасних інструментів безпеки, таких як SuperTokens, Firebase Auth, Keycloak, Supabase, Auth0, Centrifugo, Pushbullet, Ntfy та Gotify, їх роль у створенні безпечного хмарного середовища стає все більш значущою. Ці інструменти надають широкий спектр можливостей для ефективного управління ідентичністю користувачів, автентифікації та реалізації надійних механізмів сповіщень. Вони дозволяють компаніям не тільки відповідати суворим вимогам безпеки, але й ефективно адаптуватися до швидкозмінних умов, що є характерними для динамічного хмарного середовища.

Centrifugo та SuperTokens відіграють вирішальну роль у сфері веб-розробки, виконуючи специфічні та важливі функції. Centrifugo, як масштабований сервер реального часу, спеціалізується на мовно-незалежному обміні повідомленнями, рисунок 2.6. Його головна сила полягає у забезпеченні надійної та ефективної

комунікації в реальному часі, що є необхідним для сучасних веб-додатків. Це особливо важливо для функцій, таких як чати та онлайн-оновлення, де швидкий обмін інформацією є критичним [15].

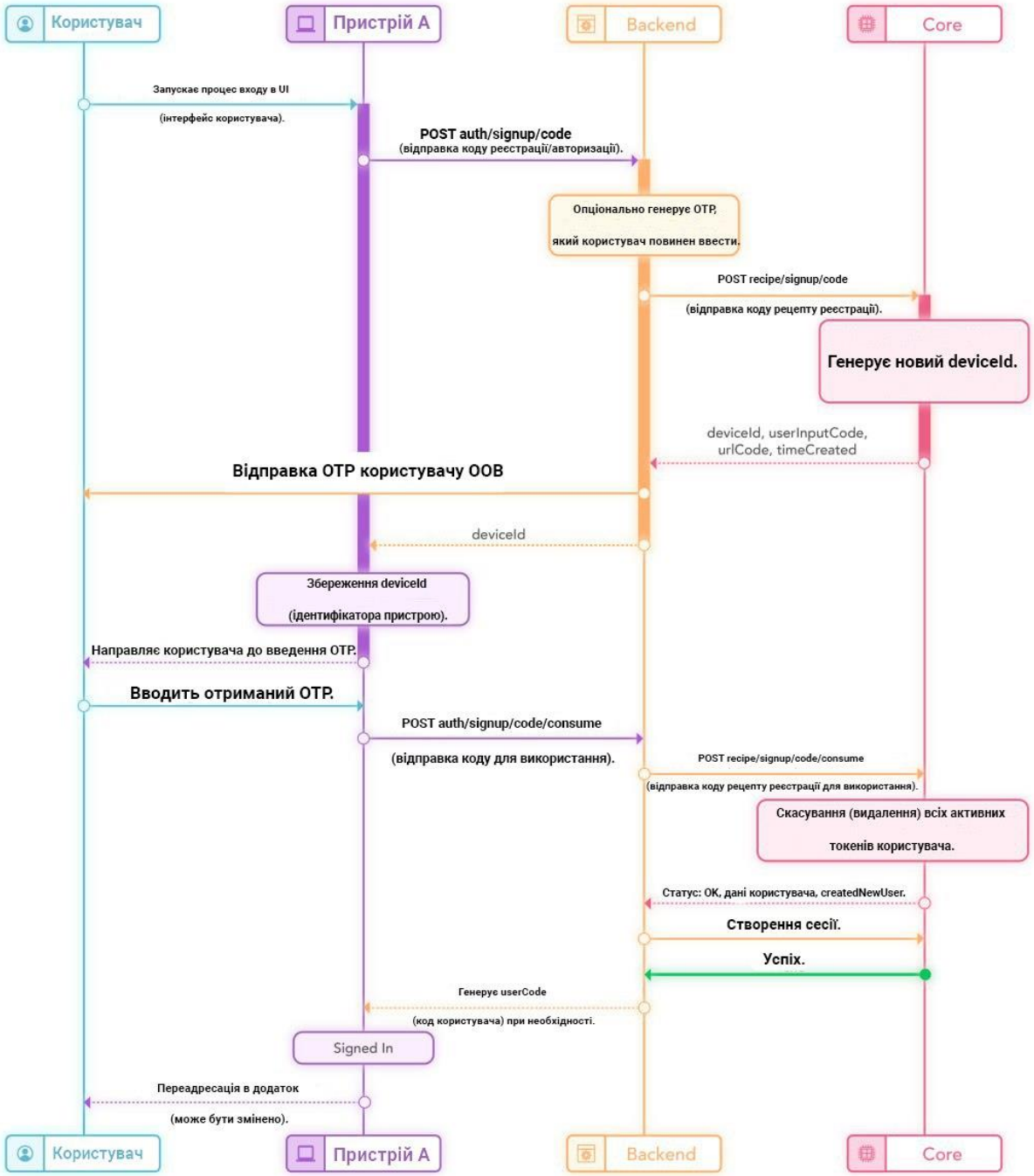


Рис. 2.6 Схема взаємодії користувача з іншими сервісами або пристроями при використанні SuperTokens

Серед альтернатив Centrifugo, Pushbullet дозволяє легко передавати контент, такий як посилання та файли, між різними пристроями. Ntfy використовується для отримання повідомлень із скриптів або додатків за допомогою простих PUT/POST запитів, забезпечуючи зручність у реалізації повідомлень. Gotify спеціалізується на push-повідомленнях через REST-API, що робить його ідеальним для середовищ на основі Linux. Pushover інтегрується з веб-додатками та системами моніторингу, забезпечуючи надійну систему сповіщення [20]. Kopy, у свою чергу, зосереджений на копіюванні тексту між пристроями, що полегшує обмін інформацією. З іншого боку, SuperTokens виступає як відкрите рішення для аутентифікації користувачів. Воно значно спрощує процес інтеграції безпечної аутентифікації в мобільні додатки та веб-сайти. Це дозволяє розробникам зосередитися на основних аспектах їх додатків, не турбуючись про складнощі впровадження систем аутентифікації. Важливими альтернативами SuperTokens є Firebase, який надає повний набір інструментів для розвитку веб-додатків, Keycloak як відкрите рішення для управління ідентичністю та доступом, Supabase, який надає реальні та RESTful API до баз даних PostgreSQL, Etebase для розробки додатків з кінцевим зашифруванням та Auth0 як універсальне рішення для управління ідентичністю.

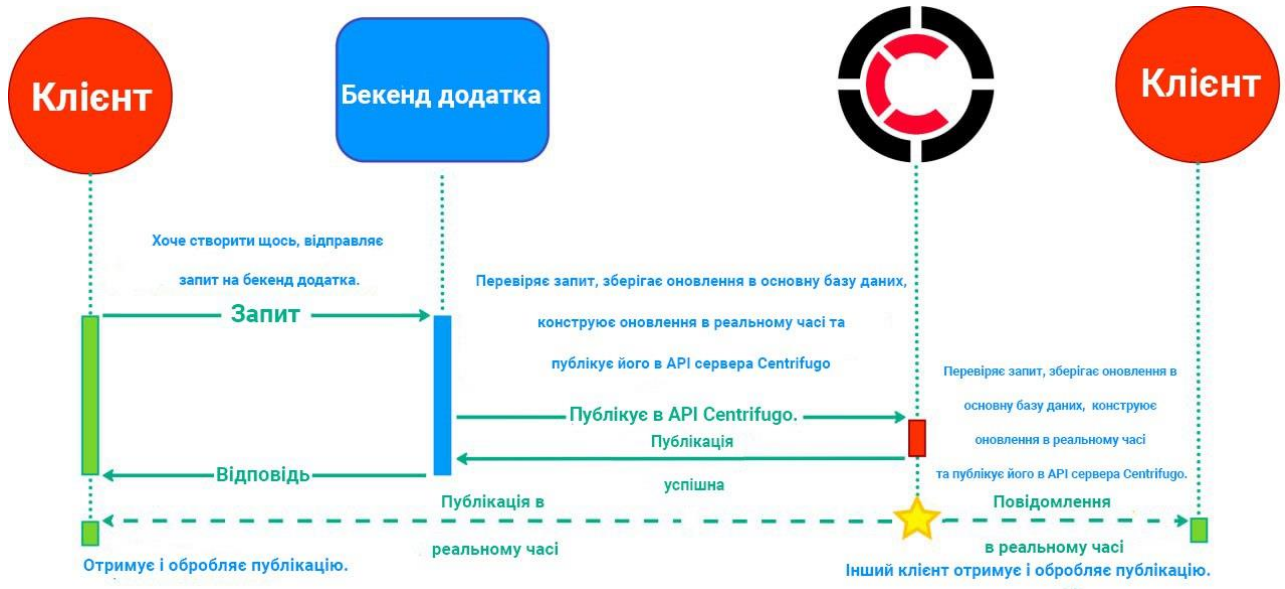


Рис. 2.7 Схема роботи Centrifugo при інтеграції в додаток

Centrifugo та SuperTokens, хоч і виконують різні функції, взаємодоповнюють один одного у процесі розробки надійних та безпечних веб- та мобільних додатків. Centrifugo вносить важливий вклад у покращення комунікації в реальному часі, що є ключовим для динамічних та взаємодіючих веб-додатків, рисунок 2.7. Тим часом, SuperTokens спеціалізується на аутентифікації та управлінні ідентичністю користувачів, забезпечуючи безпечний доступ та управління даними користувачів. Ці інструменти разом створюють більш комплексний та ефективний підхід до розробки веб- та мобільних додатків, підвищуючи їх функціональність та безпеку [14].

Інтеграція цих інструментів безпеки у хмарні платформи підкреслює готовність останніх відповідати викликам сучасного цифрового світу, де безпека та конфіденційність даних є фундаментальними вимогами. Це не лише забезпечує надійний захист важливої інформації, але й сприяє підвищенню довіри та надійності у відносинах з користувачами та клієнтами. Важливо відзначити, що в умовах постійно зростаючих загроз у сфері кібербезпеки, питання забезпечення безпеки в хмарних сховищах стає ще більш актуальним [17].

Таким чином, можна констатувати, що забезпечення безпеки даних у хмарних сховищах є ключовим аспектом розвитку хмарних технологій. Це включає не тільки застосування передових технічних рішень, але й впровадження комплексних стратегій, що враховують регуляторні вимоги та забезпечують високий рівень захисту інформації. Спільними зусиллями провідних хмарних провайдерів та розробників інструментів безпеки створюється середовище, у якому дані можуть бути зберігані та оброблені з гарантією їх цілісності та конфіденційності.

3 ІМПЛЕМЕНТАЦІЯ КОМПЛЕКСНИХ МЕТОДІВ ЗАХИСТУ ДАНИХ В ХМАРНОМУ СХОВИЩІ

3.1 Розробка системи авторизації з використанням SuperTokens

Для забезпечення безпеки в хмарному сховищі, яка пов’язана з авторизацією та аутентифікацією користувача, а також керуванням його сесій, використовуються певні інструменти. SuperTokens — це відкритий і гнучкий фреймворк, який надає розробникам набір безпечних та легко інтегрованих інструментів для побудови систем авторизації та управління сесіями користувачів. Він заснований на плагін-орієнтованій архітектурі, яка дозволяє легко додавати, вилучати та налаштовувати різноманітні аспекти аутентифікації залежно від потреб проекту.

Коли мова заходить про вибір відповідного модуля з SuperTokens, розробники повинні розглянути різні доступні опції. SuperTokens пропонує ряд модулів, таких як EmailPassword, ThirdParty, ThirdPartyEmailPassword, і Session. Кожен з цих модулів має свої особливості. Наприклад, якщо розробник хоче дозволити користувачам входити через соціальні мережі, то модуль ThirdParty буде ідеальним вибором. Водночас, якщо потрібно дозволити авторизацію як з використанням електронної пошти та пароля, так і через треті сторони, то краще вибрати модуль ThirdPartyEmailPassword. Детальний огляд кожного модуля:

– модуль EmailPassword забезпечує традиційний підхід до аутентифікації, де користувачі реєструються та входять у систему, використовуючи свою електронну адресу та пароль. Цей підхід вимагає від користувачів створення та запам'ятовування пароля, що часто може бути викликом у плані забезпечення безпеки, оскільки користувачі можуть використовувати слабкі або повторювані паролі;

– модуль ThirdParty дає змогу користувачам входити в систему за допомогою зовнішніх постачальників ідентичності, таких як Google, Facebook, або GitHub. Це

спрощує процес реєстрації, оскільки користувачам не потрібно створювати новий пароль, і може збільшити конверсію реєстрацій, завдяки скороченню кількості кроків, необхідних для входу;

– модуль ThirdPartyEmailPassword комбінує можливості двох попередніх модулів, дозволяючи користувачам реєструватися та входити в систему як з використанням електронної пошти та пароля, так і через сторонні сервіси. Це гнучкий підхід, який може задовольнити широкий спектр вимог та переваг користувачів;

– модуль Session відповідає за керування сесіями користувачів. Він використовується для відстеження стану аутентифікації користувача на сервері та в браузері, а також для захисту від загроз, таких як CSRF (Cross-Site Request Forgery) атаки.

Кожен із цих модулів вимагає від розробника ретельного налаштування та інтеграції з іншими частинами системи, включаючи фронтенд, серверну логіку, та базу даних. Застосування цих модулів дозволяє створити міцну основу для надійної та безпечної системи авторизації. В веб-додаток хмарного сховища було вирішено додати систему авторизації через email та password, тобто використати модуль EmailPassword [23].

Після вибору потрібного модуля наступним кроком є його конфігурація. Це включає в себе налаштування API ключів, встановлення параметрів безпеки, і налаштування повідомлень, які будуть відображатися користувачам під час різних етапів авторизації. Конфігурація також має на увазі встановлення параметрів cookie, часу життя сесії, та налаштування політики CORS, якщо веб-додаток має взаємодіяти з іншими доменами.

Початковий етап розробки системи авторизації з використанням — це детальне планування, яке вимагає чіткого розуміння бізнес-вимог, потреб користувачів та технічних можливостей системи. Беручі до уваги ці фактори, було розроблено алгоритм процесу авторизації користувача використовуючи SuperTokens, який інтегровано разом з бекенд ядром.

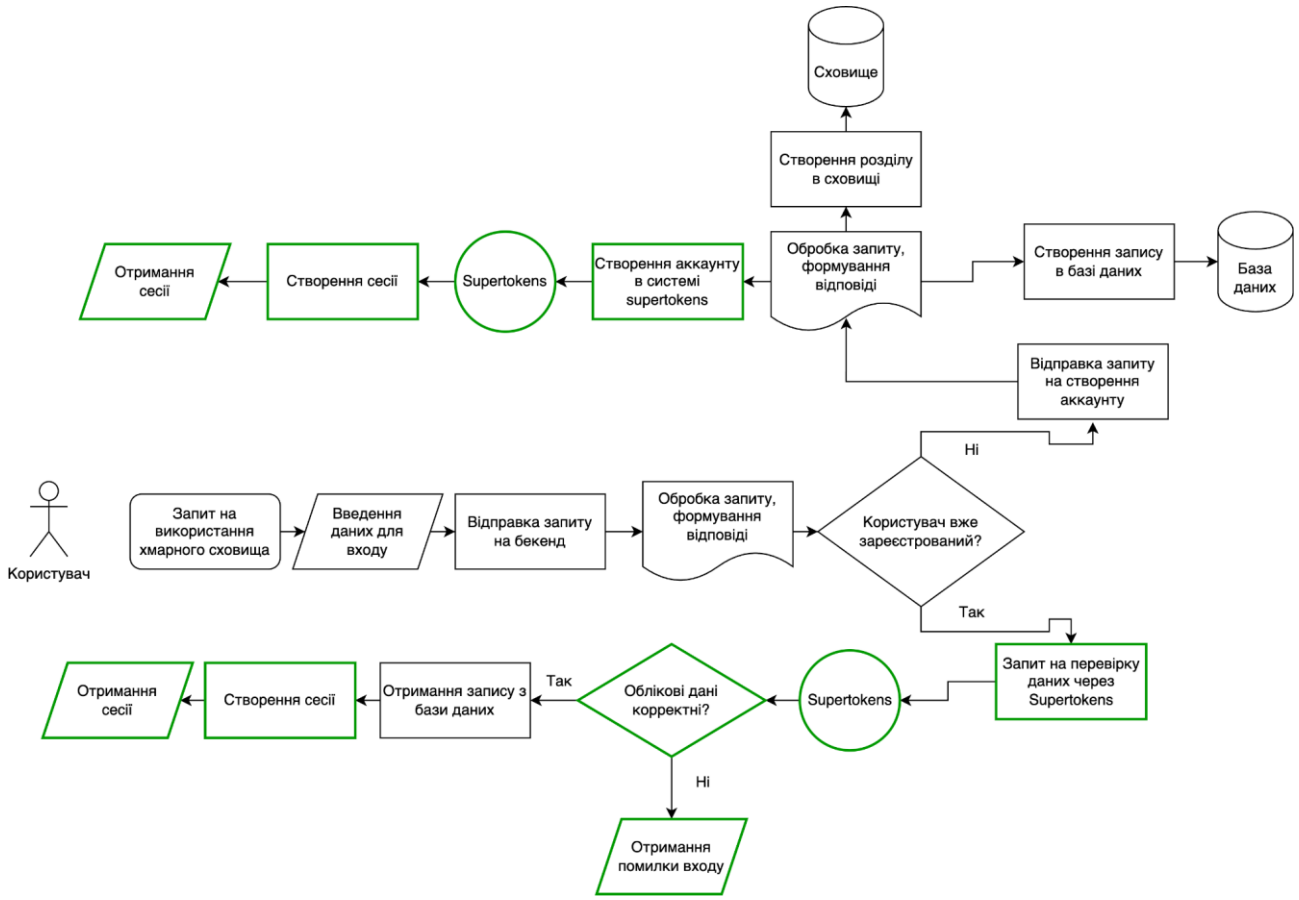


Рис. 3.1 Алгоритм процесу авторизації користувачів

На рисунку 3.1, процес починається з користувача, який намагається увійти, вводячи дані для входу. Ці дані відправляються на сервер, де відбувається їх обробка і перевірка. Якщо користувач ще не зареєстрований, автоматично створюється запис у базі даних, на створення аккаунта. Якщо дані введено правильно і користувач існує в системі, то відбувається створення сесії. В процесі створення сесії, система SuperTokens генерує спеціальні токени, які використовуються для управління сесією. Якщо ж дані введені неправильно, користувач отримує повідомлення про помилку входу. У випадку, коли користувач вже має аккаунт, система перевіряє дані через SuperTokens. На блок-схемі також присутній процес взаємодії з базою даних: створення запису, створення розділу в сховищі та інші дії, які пов'язані з аутентифікацією та сесіями користувача.

Для керування даними та сесіями користувачів в SuperTokens існує модуль Dashboard, який надає адміністраторам потужні можливості для цього. Це веб-

інтерфейс, який візуалізує управління акаунтами та сесіями, що значно спрощує моніторинг та адміністрування системи авторизації. Починаючи з функціоналу керування користувачами, Dashboard дозволяє адміністраторам переглядати список всіх зареєстрованих у системі осіб, рисунок 3.2.

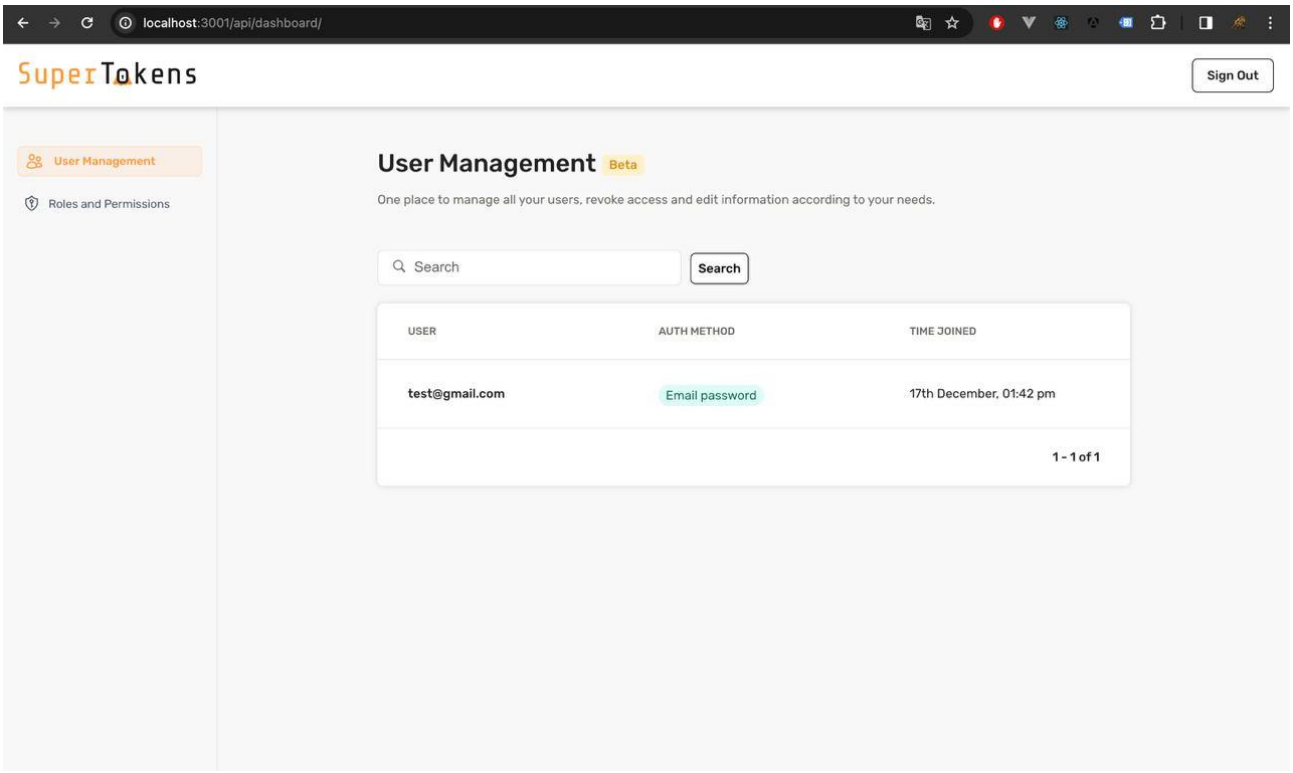


Рис. 3.2 Список зареєстрованих осіб у системі

Тут вони можуть отримати доступ до основної інформації про кожного користувача, наприклад, до їх імен, електронних адрес та статусів акаунтів. Це особливо зручно для великих систем, де кількість користувачів може сягати десятків тисяч або навіть більше. Що стосується пошуку та фільтрації, Dashboard надає інструменти для швидкого знаходження користувачів за різними параметрами. Адміністратори можуть використовувати пошук для знаходження профілів за іменем, електронною адресою або іншими визначальними характеристиками. Фільтри допомагають вузько спеціалізувати перегляд, відокремлюючи, наприклад, тільки тих користувачів, які чекають на підтвердження електронної пошти або тих, хто нещодавно змінив свій пароль.

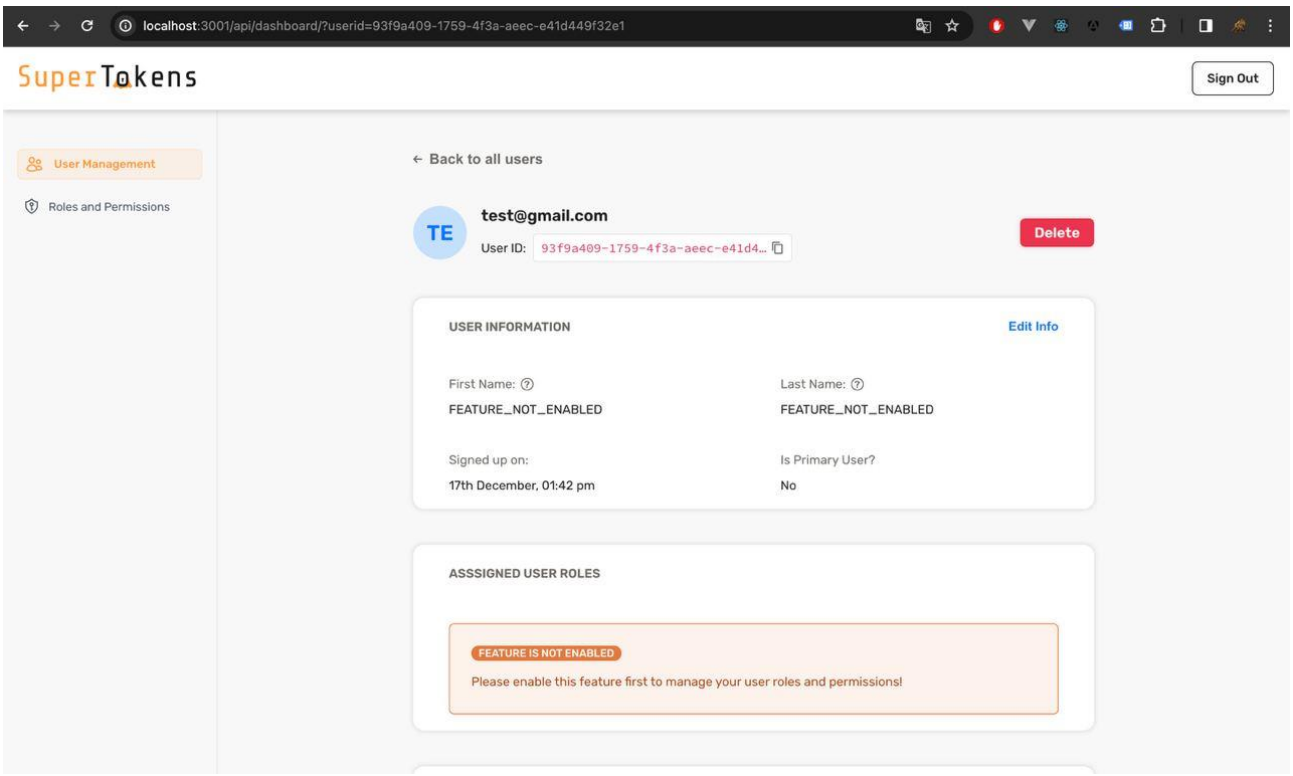


Рис. 3.3 Дані про користувача в системі

У сфері редагування профілів, Dashboard надає змогу змінювати особисті дані користувачів, їх ролі в системі, і навіть статуси їх акаунтів, рисунок 3.3. Це може бути корисним у випадках, коли потрібно надати користувачу додаткові права доступу або, навпаки, обмежити його можливості через порушення правил користування сервісом. Блокування та розблокування акаунтів є ще однією критичною функцією, яку надає Dashboard. У випадках підозрілих діяльностей або порушень, адміністратор може швидко втрутитись, блокуючи доступ до системи для певних акаунтів. З іншого боку, якщо ситуація була вирішена, доступ можна легко відновити.

Переходячи до управління сесіями, Dashboard відображає всі активні сесії, надаючи детальну інформацію про час створення та закінчення кожної з них.

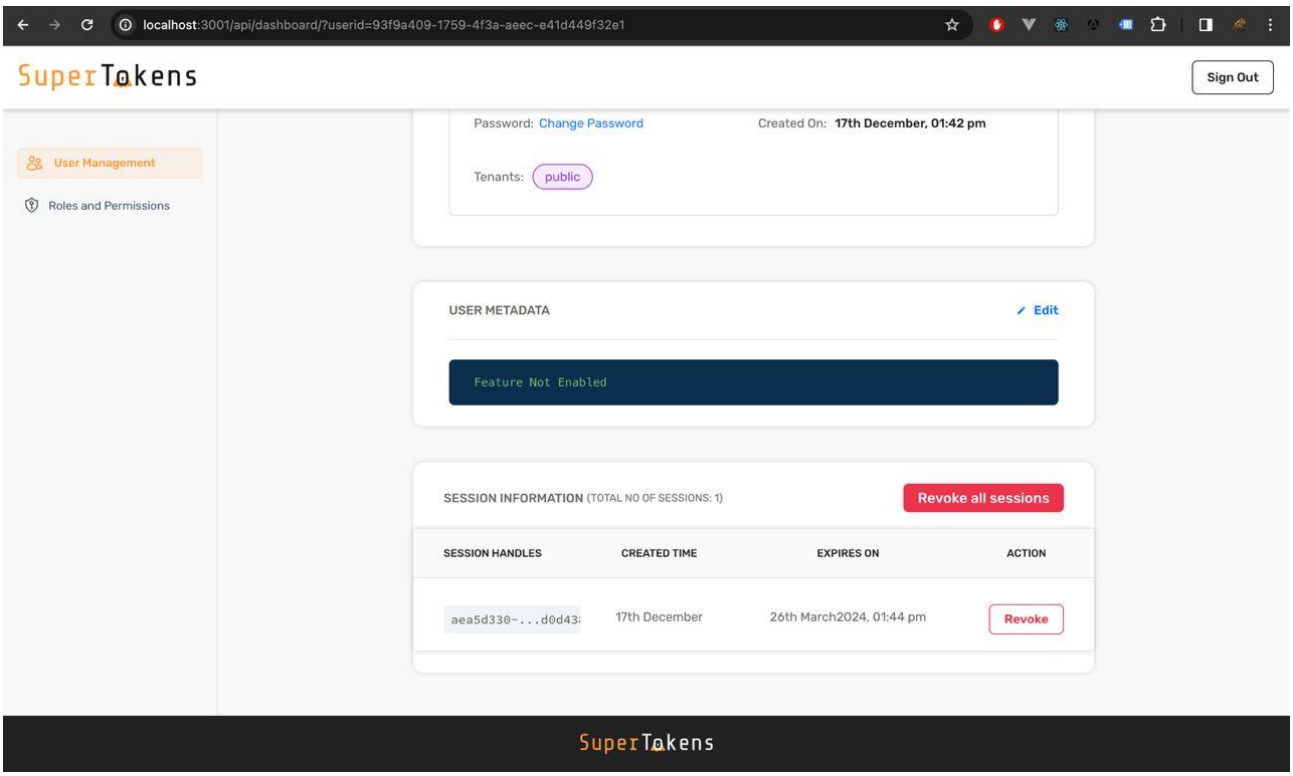


Рис. 3.4 Керування сесіями користувача

Це дає змогу адміністраторам моніторити активність користувачів та виявляти нестандартні ситуації, такі як підозріла кількість одночасних сесій або сесії з несподіваних географічних локацій. Крім того, Dashboard дозволяє адміністраторам вручну закривати будь-яку сесію, рисунок 3.4. Це може бути корисно у випадку виявлення безпекових порушень або якщо користувач забув вийти із системи на публічному пристрої. Завершення сесії може бути також застосоване як частина процедури відновлення акаунту після злому. Резюмуючи, Dashboard від SuperTokens — це місце централізованого контролю за користувачами та їх сесіями, яке забезпечує адміністраторам гнучкість, ефективність та високий рівень безпеки. Цей інструмент є незамінним для керування великими користувацькими базами, забезпечуючи зручний інтерфейс для виконання адміністративних завдань без прямого доступу до бази даних чи бекенд-логіки.

Беручі до уваги, ці фактори та можливості, було розроблено алгоритм процесу керування сесіями користувача.

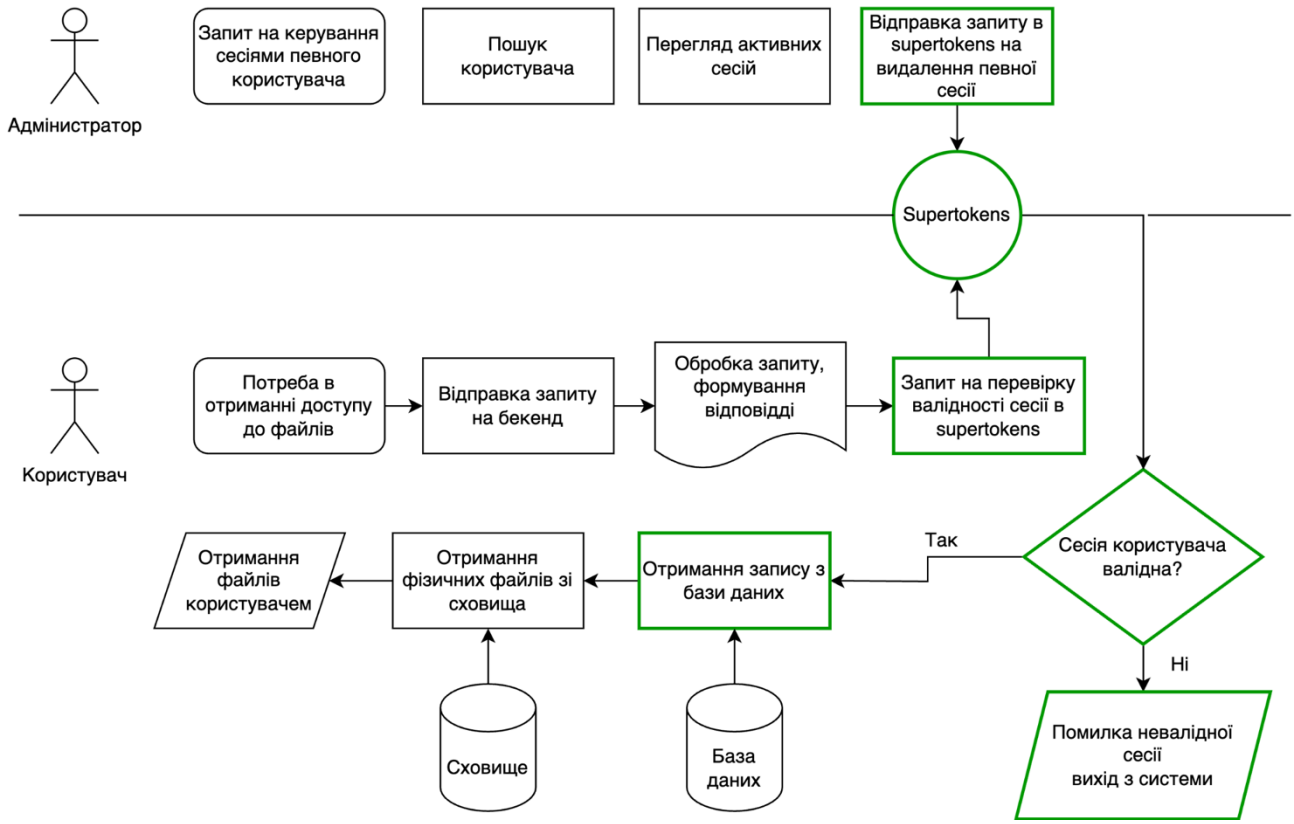


Рис. 3.5 Алгоритм процесу керування сесіями користувача

На рисунку 3.5, з лівого верхнього кута блок-схеми починається дія адміністратора, який здійснює запит на керування сесіями певного користувача. Далі адміністратор може провести пошук користувача і переглянути активні сесії. Якщо потрібно, адміністратор відправляє запит в систему SuperTokens для виділення певної сесії. На нижній частині блок-схеми зображений користувач, який має потребу в отриманні доступу до файлів. Користувач відправляє запит на бекенд, після чого система обробляє запит та формує відповідь. Якщо сесія користувача валідна, як це визначається через SuperTokens, користувач отримує доступ до файлів. Це може включати отримання фізичних файлів із сховища або записів з бази даних. Якщо ж сесія користувача не валідна, то користувач отримує повідомлення про помилку невалідної сесії та вихід з системи. Таким чином, ця блок-схема відображає потоки робочих процесів управління доступом до ресурсів у системі та взаємодію між адміністратором, користувачем і системою аутентифікації.

Використовуючи дані інструменти для розробки систем авторизації та управління сесіями користувачів у хмарних сховищах, підвищується безпека, яка дозволяє гнучко налаштовувати аутентифікацію залежно від вимог проекту. Різноманітність модулів, таких як EmailPassword для традиційних методів входу, ThirdParty для аутентифікації через соціальні мережі, та інших, разом із Dashboard для керування даними користувачів, забезпечують повну контрольність та високий рівень безпеки, що є критично важливим для хмарних платформ.

3.2 Застосування Centrifuge.js для забезпечення взаємодії в реальному часі

Для забезпечення безпеки в хмарному сховищі, яка пов’язана з взаємодією фронтенд та бекенд частини в реальному часі, використовуються певні інструменти. Centrifuge.js є потужною бібліотекою для реалізації взаємодії в реальному часі в веб-додатках. Вона використовується для створення швидкого та надійного каналу комунікації між сервером та клієнтами через WebSockets, що забезпечує миттєвий обмін повідомленнями та даними. Це особливо важливо у сучасних веб-додатках, де користувачі очікують негайних оновлень та інтерактивності без необхідності перезавантажувати сторінку. У веб-додатку хмарного сховища Centrifuge.js виконує ключову роль, діючи як "безпековий коридор" для взаємодії між клієнтом і сервером. Коли користувач виявляє потребу в завантаженні файлу, він ініціює процес, натискаючи на інтерфейс завантаження. Запит тоді відправляється на бекенд через безпечне з'єднання. Сервер обробляє запит, зберігає фізичний файл у сховищі і створює відповідний запис у базі даних. Після завантаження файлу, важливо, щоб користувачі могли отримати оновлення в реальному часі. Це може бути підтвердження успішного завантаження або надсилання оновлень щодо статусу процесу. Ось тут і знадобиться Centrifuge.js, яка публікує відповіді на сервері, які відразу передаються клієнтам через вебсокети. Наприклад, якщо в бекенді виникає подія, що вимагає уваги користувача, така як завершення завантаження файлу, ця

інформація може бути миттєво опублікована у відповідний канал у Centrifuge.js, і всі підписані на цей канал користувачі отримають повідомлення.

Centrifuge.js також підвищує безпеку завдяки можливості використання токенів аутентифікації та шифрування з'єднань. Кожен канал може бути захищений токенами, що забезпечують, що тільки авторизовані користувачі можуть підписатися на оновлення. Це важливо для захисту конфіденційності та інтегритету даних, які передаються. Хмарне сховище може використовувати Centrifuge.js для створення надійної та ефективної системи обміну даними, яка не тільки забезпечує взаємодію в реальному часі, але й дотримується високих стандартів безпеки. Через це користувачі можуть бути впевнені, що їх дані обробляються конфіденційно та що вони миттєво отримують оновлення, необхідні для забезпечення продуктивної роботи.

Беручі до уваги, ці фактори та можливості, було розроблено алгоритм процесу отримання даних по захищеному каналу

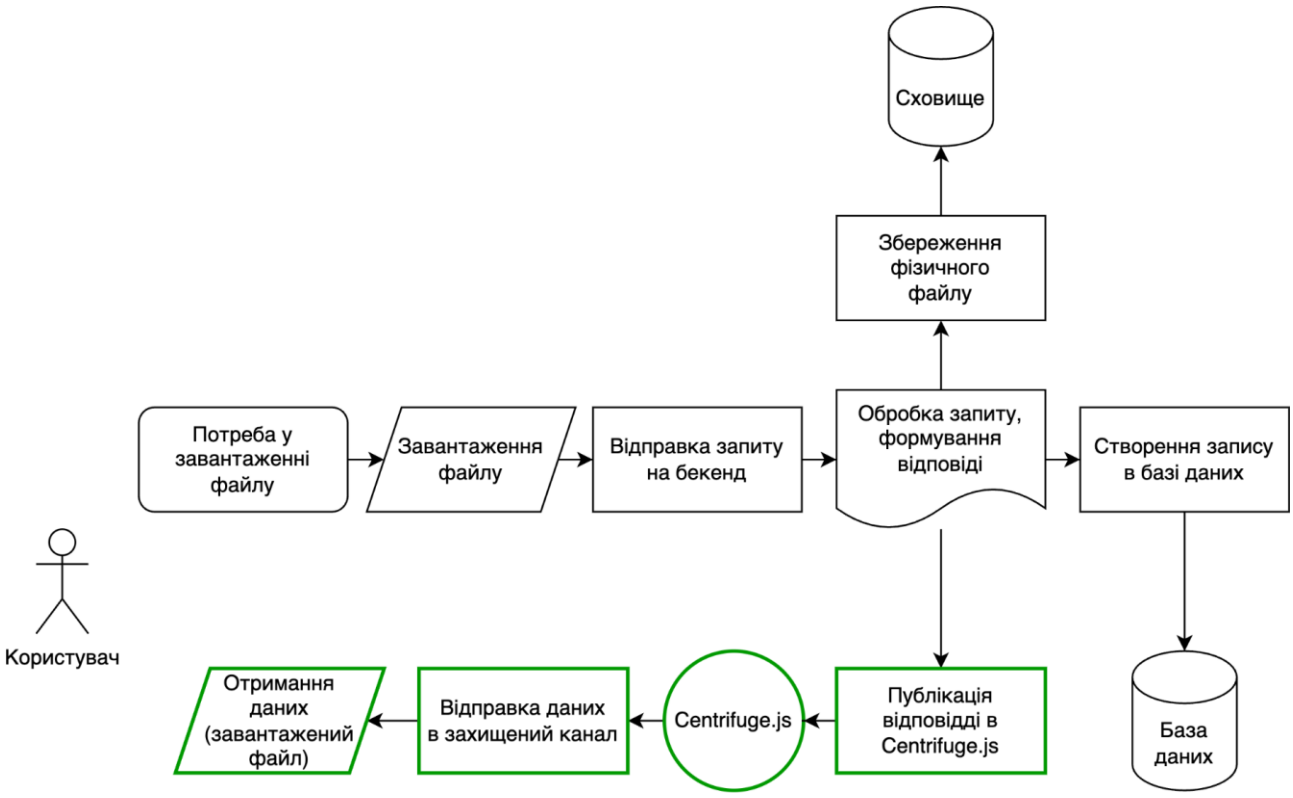


Рис. 3.6 Алгоритм процесу отримання даних по захищеному каналу

На рисунку 3.6, алгоритм ілюструє процес завантаження файлу користувачем та його подальшу обробку на сервері з використанням Centrifuge.js. Процес починається з користувача, який має потребу у завантаженні файлу. Користувач завантажує файл, який потім відправляється на бекенд сервера. На сервері відбувається обробка запиту і формування відповіді. Як частина цієї обробки, файл фізично зберігається в сховищі, а інформація про файл записується в базу даних. Далі, відповідь публікується через Centrifuge.js, який є реалізацією WebSocket або подібного протоколу для забезпечення двостороннього з'єднання між клієнтом і сервером в реальному часі. Це дозволяє користувачу отримувати дані (завантажений файл) через захищений канал, забезпечуючи швидке і ефективне спілкування. Таким чином, блок-схема відображає повний цикл від моменту, коли користувач виявляє потребу в завантаженні файлу, до отримання цього файлу через сучасний механізм реалізований за допомогою Centrifuge.js.

Розглянемо інтерфейс взаємодії з Centrifuge.js, централізованим місцем для керування взаємодією в реальному часі між сервером і клієнтами. Цей інтерфейс, як правило, має інтуїтивно зрозумілий графічний користувацький інтерфейс, який дозволяє адміністраторам легко виконувати наступні дії:

– відправлення повідомлень: Адміністратори можуть публікувати повідомлення в різні канали, використовуючи просту форму введення, де вони можуть вказати канал та вміст повідомлення у форматі JSON;

– моніторинг статусу: Dashboard може відображати поточний статус сервера Centrifuge.js, включаючи активні з'єднання, кількість повідомлень, що обробляються, та інші метрики працездатності;

– конфігурація каналів: Інтерфейс надає можливість налаштувати властивості каналів, включаючи приватність, токени доступу та параметри шифрування;

– логування дій: Адміністратори можуть переглядати логи дій, які включають історію всіх команд, відправлених через Dashboard, та відповіді від сервера;

– управління підписками: Dashboard дозволяє адміністраторам керувати підписками користувачів на канали, дозволяючи їм вручну додавати або видаляти підписки для конкретних користувачів;

– тестування функцій: Часто такі Dashboard містять інструменти для тестування різних сценаріїв взаємодії в реальному часі, щоб переконатися у правильній роботі системи перед її запуском у виробництво.

Інтерфейс зазвичай доповнюється додатковими утилітами для імпорту чи експорту конфігурацій, управління версіями API та інтеграції з зовнішніми системами моніторингу та сповіщень.

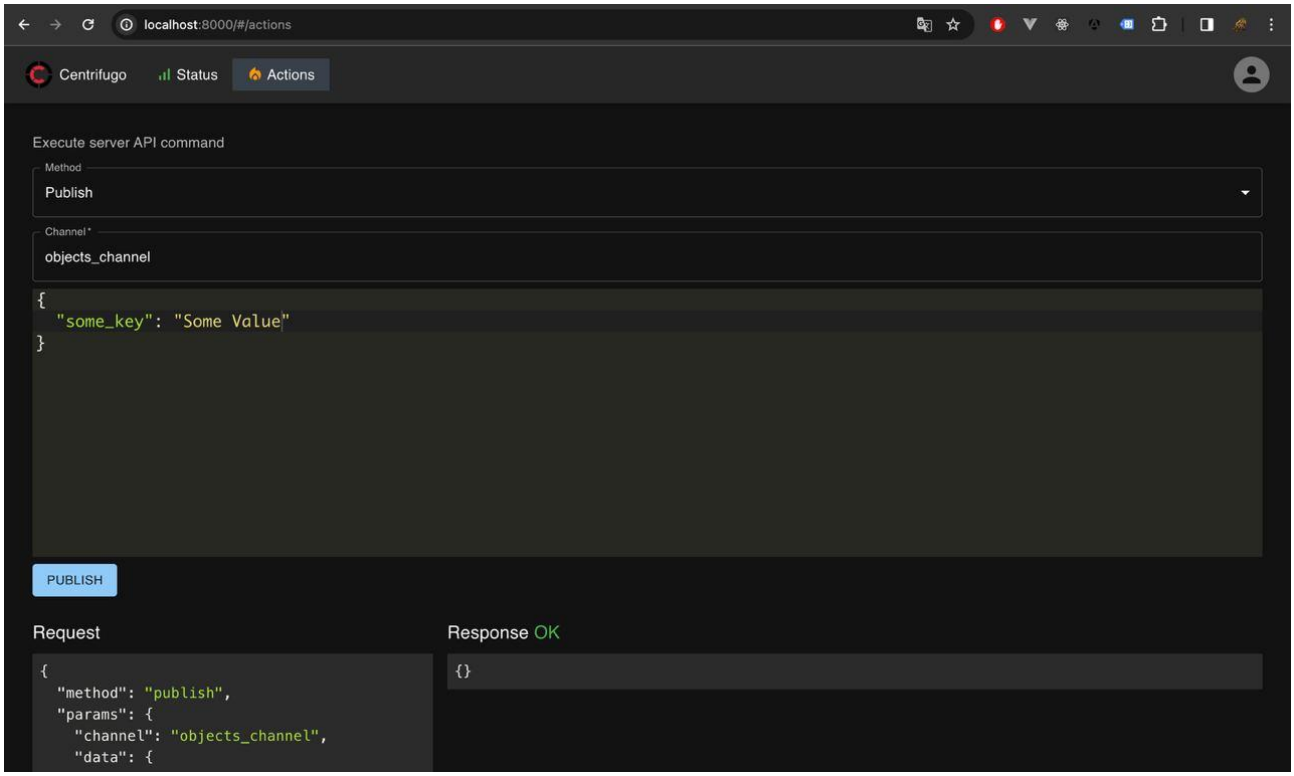


Рис. 3.7 Приклад відправки повідомлень

На рисунку 3.7, процес починається, коли адміністратор або система вводить команду в інтерфейс, вказуючи метод publish для відправлення даних. У полі "Channel" вводиться назва каналу, у який будуть надсилатися дані, у цьому випадку це "objects_channel". У вікні з JSON-структурою вводяться дані, які необхідно відправити, наприклад { "some_key": "Some Value" }. Ці дані визначають вміст повідомлення, яке буде опубліковано. Після натискання кнопки "PUBLISH", Dashboard відправляє запит до сервера Centrifuge.js. Запит включає метод, назву каналу та дані, які потрібно передати. Сервер обробляє цей запит, публікує дані у вказаний канал і надсилає відповідь назад до Dashboard, яка

підтверджує успішну публікацію з відповіддю "Response OK". Це показує, як адміністратори можуть інтерактивно керувати взаємодією в реальному часі, використовуючи Centrifuge.js для надсилання повідомлень та даних у конкретні канали, до яких підключені клієнти. Цей ефективний спосіб для реалізації функціоналу подій в реальному часі, таких як сповіщення користувачів, синхронізація стану в багатокористувацьких додатках, або надсилання оновлень у реальному часі.

3.3 Аналіз та оптимізація ефективності системи захисту

Оцінюючи систему до та після впровадження методів захисту, можна отримати чітке уявлення про ефективність впроваджених змін. Ці зміни можуть бути аналізовані з допомогою кількісних показників, таких як час відгуку системи, пропускна здатність під час пікових навантажень, стабільність роботи системи, а також через кількісну оцінку витрат на підтримку. Такий підхід дозволяє не тільки виявити слабкі місця в архітектурі та стратегії безпеки, але й підтвердити виправданість інвестицій в новітні технології та рішення, які впроваджуються для підвищення масштабованості та надійності системи.

Такий комплексний аналіз дозволяє отримати повне і чітке уявлення про вплив цих заходів на загальну продуктивність та ефективність системи. Аналізуючи ключові кількісні показники, такі як час відгуку системи, ми можемо визначити, як швидко система реагує на запити користувачів, що є критичним для забезпечення високого рівня користувацького досвіду. Пропускна здатність під час пікових навантажень вказує на здатність системи обробляти велику кількість запитів одночасно, що є важливим для розуміння меж системи та її потенціалу до масштабування під час сплесків активності.

Подальша оцінка включає аналіз технологій та протоколів аутентифікації та авторизації, які використовуються системою. Сучасні рішення, такі як JWT, OAuth 2.0, та OpenID Connect, забезпечують вищий рівень безпеки порівняно з застарілими методами, такими як базова аутентифікація через cookies та сесії.

Вони використовують складні механізми для забезпечення безпеки токенів та даних, що дозволяє підвищити довіру користувачів та виконувати регуляторні вимоги.

Останній, але не менш важливий аспект оцінки — це рівень масштабованості системи. Здатність системи адаптуватися до зростаючих потреб та збільшення обсягів даних, не втрачаючи при цьому продуктивності, є ключовим фактором, який впливає на довготривалу стійкість бізнесу в динамічному цифровому середовищі. Системи, які можуть масштабуватися, дозволяють бізнесу рости без необхідності постійного втручання в інфраструктуру.

Таблиця 3.1

Оцінка масштабованості та надійності системи

Показник	**До впровадження методів захисту**	**Після впровадження методів захисту**
Час відгуку системи (мс)	500	100
Пропускна здатність під час пікових навантажень (запитів/сек)	50	300
Стабільність під час пікових навантажень	часткові збої	стабільна
Витрати на підтримку	високі	нижчі
Технології та протоколи аутентифікації та авторизації	Cookies і Сесії, Basic Authentication	JWT, OAuth 2.0, OpenID Connect
Рівень масштабованості системи	Низький	високий

В таблиці 3.1, оцінка масштабованості та надійності системи може бути проведена на основі різних технічних показників, зображених у таблиці. Для

цього можемо використати математичний підхід, аналізуючи дані до та після впровадження методів захисту.

Час відгуку системи — це період часу між надсиланням запиту користувачем та отриманням відповіді від системи. Цей показник важливий, оскільки він впливає на сприйняття користувачем швидкості та ефективності системи. В ідеальному випадку час відгуку має бути якомога меншим, оскільки це сприяє кращому користувацькому досвіду. Якщо розглянути приклад з наведеної таблиці, де час відгуку системи був зменшений з 500 мс до 100 мс після впровадження методів захисту, це можна кількісно оцінити за допомогою простих математичних розрахунків.

Нехай $L_{\text{базовий}}$ — базовий час відгуку системи без будь-яких покращень, а D — час затримки, який додається через недоліки у старій системі. Тоді старий час відгуку системи ($L_{\text{старий}}$) можна виразити як суму базового часу відгуку і доданої затримки, вираз (3.1):

$$L_{\text{старий}} = L_{\text{базовий}} + D \tag{3.1}$$

Після впровадження покращень, додана затримка D зменшується завдяки оптимізації системи. Якщо впровадження покращень зменшує затримку в чотири рази, тоді новий час відгуку ($L_{\text{новий}}$) буде таким, вираз (3.2):

$$L_{\text{новий}} = L_{\text{базовий}} + \frac{D}{4} \tag{3.2}$$

Пропускна здатність системи під час пікових навантажень — це вимір того, скільки запитів система може ефективно обробити за одну секунду. Цей показник є критичним для веб-сервісів та інтерактивних додатків, де висока пропускна здатність забезпечує плавність та відсутність затримок при високому об'ємі користувацьких запитів. Чим вища пропускна здатність, тим краще система може справлятися з раптовими сплесками активності, такими як онлайн-продажі або події, що залучають велику аудиторію. Якщо взяти до уваги дані з таблиці, то ми бачимо, що пропускна здатність системи зросла з 50 запитів на секунду до 300 запитів на секунду після впровадження покращень.

Нехай наша формула має такі вхідні дані: P — пропускна здатність системи (запитів/сек), K — коефіцієнт ефективності системи, M — максимально можлива пропускна здатність системи при оптимальних умовах. За умови, що M є сталою величиною, ми можемо розглянути зміну K для отримання різних значень P. Нехай $M = 1500$ (припустимо, що це максимально можлива пропускна здатність системи). Тоді для старої пропускної здатності, вирази (3.3), (3.4), (3.5):

$$50 = K \times 1500 \tag{3.3}$$

$$K = \frac{50}{1500} \tag{3.4}$$

$$K = \frac{1}{30} \tag{3.5}$$

Тепер нехай покращення системи збільшили K . Для нової пропускної здатності, вирази (3.6), (3.7), (3.8):

$$300 = K \times 1500 \tag{3.6}$$

$$K = \frac{300}{1500} \tag{3.7}$$

$$K = \frac{1}{5} \tag{3.8}$$

Стабільність системи під час пікових навантажень — це ключовий показник, який вказує на здатність системи підтримувати надійну та безперебійну роботу навіть у моменти високої активності та великого обсягу запитів. Цей аспект важливий, оскільки він безпосередньо впливає на довіру користувачів та загальну задоволеність ними послугами системи. Для ілюстрації, розглянемо ситуацію, де система до впровадження покращень мала певний рівень стабільності, наприклад, вона мала часткові збої або уповільнення під час пікових навантажень. Після впровадження заходів щодо оптимізації та підсилення інфраструктури, система стала працювати стабільно навіть у періоди високої активності.

Математично це можна виміряти за допомогою показника Mean Time Between Failures (MTBF), який вказує на середній час між збоями системи. Нехай MTBF до покращень був X годин, а після покращень — значно збільшився. Розрахунок MTBF виглядає так, вираз (3.9):

$$MTBF = \frac{\text{Загальний час роботи системи}}{\text{Кількість збоїв}} \quad (3.9)$$

Припустимо, система працювала 1200 годин і мала 10 збоїв до покращень, тоді вираз (3.10) буде:

$$MTBF_{\text{до}} = \frac{1200}{10} = 120 \text{ годин} \quad (3.10)$$

Після покращень кількість збоїв значно зменшилася, скажімо до 2 збоїв за той самий час, тоді вираз (3.11) буде:

$$MTBF_{\text{після}} = \frac{1200}{2} = 600 \text{ годин} \quad (3.11)$$

Витрати на підтримку системи — це сума ресурсів, які необхідні для її обслуговування та забезпечення належного функціонування. Це включає технічну підтримку, оновлення програмного забезпечення, моніторинг системи, управління інфраструктурою та інші витрати, пов'язані з утриманням системи в робочому стані. Ці витрати важливі, оскільки вони впливають на загальну рентабельність системи та її ефективність. В ідеальному випадку витрати на підтримку мають бути оптимізовані, щоб забезпечити максимальну продуктивність системи за мінімальних витрат. Розглянемо приклад, де витрати на підтримку системи були знижені після впровадження певних методів оптимізації. Це можна кількісно оцінити, порівнявши загальні витрати на підтримку до та після впровадження змін.

Нехай E — це коефіцієнт ефективності покращень, де значення 1 означає повне збереження первісних витрат (без покращень), а значення 0.7 означає, що витрати були знижені на 30% завдяки покращенням, вираз (3.12).

$$V_{\text{після}} = V_{\text{до}} \times E \quad (3.12)$$

Де $V_{\text{до}}$ — первісні витрати на підтримку системи, $V_{\text{після}}$ — витрати на підтримку системи після покращень, E — коефіцієнт ефективності покращень. Якщо первісні витрати на підтримку системи ($V_{\text{до}}$) становили 100,000 доларів на

рік, і покращення знизили витрати на 30% ($E = 0.7$), тоді витрати після покращень ($V_{\text{після}}$) будуть вирази (3.13), (3.14):

$$V_{\text{після}} = 100{,}000 \times 0.7 \tag{3.13}$$

$$V_{\text{після}} = 70{,}000 \text{ доларів на рік} \tag{14}$$

Рівень масштабованості системи відіграє ключову роль у її здатності адаптуватися до зростаючих вимог і обсягів даних. Цей показник визначає, наскільки легко систему можна розширювати або модифікувати для вирішення збільшення кількості користувачів, запитів, або обробки даних. Важливість масштабованості полягає в забезпеченні стабільності системи навіть під час високих пікових навантажень та швидкому реагуванні на зміни у потребах ринку. Математично масштабованість можна оцінити за допомогою показника, як-от "коефіцієнт масштабованості". Нехай до впровадження покращень система мала коефіцієнт масштабованості $M_{\text{до}}$, а після впровадження покращень - $M_{\text{після}}$. Цей коефіцієнт може відображати, наприклад, відношення максимально можливої кількості одночасних користувачів до фактичної кількості, яку система може обслуговувати без збоїв, вираз (3.15).

$$M = \frac{\text{фактична кількість користувачів}}{\text{максимально можлива кількість користувачів}} \tag{3.15}$$

Якщо до впровадження покращень система могла обслуговувати 1000 користувачів при максимальній можливості 2000, а після покращень - 4000 користувачів при тій же максимальній можливості, вирази (3.16), (3.17):

$$M_{\text{до}} = \frac{1000}{2000} = 0.5 \tag{3.16}$$

$$M_{\text{після}} = \frac{4000}{2000} = 2 \tag{3.17}$$

Оцінюючи різні аспекти системи до та після впровадження методів захисту та оптимізації, можна зробити висновок, що ці зміни істотно покращили її загальну продуктивність та ефективність. Значне зниження часу відгуку системи з 500 мс до 100 мс, як і ріст пропускної здатності з 50 до 300 запитів на секунду,

свідчать про підвищену продуктивність та кращу спроможність справлятися з великими обсягами даних. Покращення стабільності системи під час пікових навантажень також важливе, оскільки це забезпечує надійну та безперебійну роботу, що є критичним для забезпечення високого рівня задоволеності користувачів. Зменшення витрат на підтримку свідчить про збільшену оперативну ефективність і може вказувати на успішну автоматизацію процесів чи впровадження більш ефективних технічних рішень. Водночас, впровадження сучасних технологій та протоколів аутентифікації та авторизації, таких як JWT, OAuth 2.0, та OpenID Connect, підвищують рівень безпеки системи та її відповідність сучасним стандартам.

Нарешті, значне покращення рівня масштабованості системи є ключовим для підтримання її стабільності та ефективності у відповідь на зростаючі потреби та обсяги даних. Це підтверджує, що система не тільки готова до поточних викликів, але й має потенціал для подальшого розширення та адаптації до майбутніх потреб. Усі ці фактори разом формують зрілу, надійну та масштабовану систему, здатну задовольнити вимоги сучасного динамічного цифрового середовища.

ВИСНОВОК

В результаті виконання магістерської роботи було досліджено архітектуру хмарних сховищ, їхню ефективність та безпеку. При виконанні роботи було виконано наступні задачі:

1. В результаті аналізу архітектури хмарного сховища було виявлено, що сучасні хмарні сховища надають ефективне, гнучке, та масштабоване зберігання даних. Особлива увага була приділена складовим елементам архітектури, таким як сервери даних, системи управління даними та мережеві інфраструктури, що забезпечують оптимальну працездатність та безпеку даних.
2. Дослідження показало, що процедура реєстрації та аутентифікації в хмарних сховищах є ключовою для забезпечення безпеки даних. Багаторівнева аутентифікація та використання сучасних методів верифікації користувачів значно знижують ризик несанкціонованого доступу до даних.
3. В роботі було розроблено нові стратегії шифрування та методи захисту даних, які підвищують рівень безпеки в хмарних сховищах. Ці методи включають в себе передові техніки шифрування, які забезпечують цілісність та конфіденційність даних.
4. Оцінка показала, що хмарні сховища здатні ефективно масштабуватися для відповіді на зростаючі потреби в обробці та зберіганні даних. Надійність системи забезпечується через використання резервного копіювання, відновлення даних та різноманітні механізми відновлення після збоїв.
5. Практична реалізація теоретичних концепцій у вигляді демонстраційного додатку дозволила продемонструвати ефективність запропонованих рішень. Це підтвердило важливість інтеграції теоретичних знань з практичним застосуванням в реальних умовах.

В дипломній роботі було проаналізовано та виконано всі цілі, які були поставлені на початку даної роботи.

ПЕРЕЛІК ПОСИЛАНЬ

1. Components of Cloud Computing Architecture [Електронний ресурс]. – Режим доступу: https://www.upgrad.com/blog/components-of-cloud-computing-architecture/
2. Build Apps with These Cloud Architecture Diagram Examples [Електронний ресурс]. – Режим доступу: https://www.techtarget.com/searchcloudcomputing/tip/Build-apps-with-these-cloud-architecture-diagram-examples
3. Cloud Architecture Diagrams [Електронний ресурс]. – Режим доступу: https://www.gliffy.com/resources/cloud-architecture-diagrams
4. Top 8 Cloud Computing Architecture Diagrams [Електронний ресурс]. – Режим доступу: https://online.visual-paradigm.com/knowledge/cloud-architecture-diagrams/top-8-cloud-computing-architecture-diagrams/
5. Cloud Encryption: What It Is and How It Works [Електронний ресурс]. – Режим доступу: https://www.crowdstrike.com/cybersecurity-101/cloud-security/cloud-encryption/
6. SHA256 Overview [Електронний ресурс]. – Режим доступу: https://debugpointer.com/security/sha256-overview
7. What Happens in a TLS Handshake [Електронний ресурс]. – Режим доступу: https://www.cloudflare.com/learning/ssl/what-happens-in-a-tls-handshake/
8. Cryptography: RSA Example [Електронний ресурс]. – Режим доступу: https://www.practicalnetworking.net/series/cryptography/rsa-example/
9. What is AES Encryption and How Does it Work? [Електронний ресурс]. – Режим доступу: https://www.cloudwards.net/what-is-aes/
10. Authentication, Authorization, and Identification [Електронний ресурс]. – Режим доступу: https://training.qatestlab.com/blog/technical-articles/authentication-authorization-and-identification/
11. Samani, Raj; Honan, Brian; Reavis, Jim. CSA Guide to Cloud Computing: Implementing Cloud Privacy and Security. Waltham, MA: Syngress, ©2015.

12. Vora, Zeal. Enterprise Cloud Security and Governance: Efficiently Set Data Protection and Privacy Principles. Packt Publishing - ebooks Account, 2017.

13. Anthony, Albert. Mastering AWS Security: Create and maintain a secure cloud ecosystem. ISBN: 9781788293723.

14. Centrifugo Design Overview [Електронний ресурс]. – Режим доступу: https://centrifugal.dev/docs/getting-started/design

15. SuperTokens Flow Diagrams [Електронний ресурс]. – Режим доступу: https://supertokens.com/docs/passwordless/flow_diagram

16. AWS Documentation [Електронний ресурс]. – Режим доступу: https://docs.aws.amazon.com/

17. Google Cloud Documentation [Електронний ресурс]. – Режим доступу: https://cloud.google.com/docs

18. Microsoft Azure Documentation [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/en-us/azure/?product=popular

19. IBM Cloud Documentation [Електронний ресурс]. – Режим доступу: https://cloud.ibm.com/docs

20. Oracle Cloud Documentation [Електронний ресурс]. – Режим доступу: https://docs.oracle.com/en/cloud/get-started/index.html

21. Мардер Н.С. Сучасні телекомунікації. Вид-во IPIAC, 2006. 384 с. О`Коннор Дж. Мистецтво системного мислення. Вид-во Бібліотека - MTI, 2006. 127 с.

22. Калеб Д.. Представляємо Go. Вид-во O'Reilly Media, 2016. 128 с.

23. Машинне навчання з Go: реалізуйте регресію / Деніел У. – Вид-во Packt, 2017. – 499 с.

24. Інтеграція штучного інтелекту для підвищення захисту даних хмарного сховища / Мислюк А.С., Бондарчук А.П. // Науково-практична конференція «Проблеми Комп'ютерної Інженерії», збірник тез. – 1 грудня 2023 року, Державний Університет Інформаційно-Комунікаційних Технологій, Київ, Україна. – С. 32.

25. Сучасні методи захисту даних у сфері фінтех: інноваційні підходи та їх ефективність / Мислюк А.С., Бондарчук А.П. // VI Міжнародна наукова

конференція «Науковий простір: актуальні питання, досягнення та інновації», збірник тез. – 15 грудня 2023 року, м. Київ, Україна. – С. 363.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Магістерська робота

«РОЗРОБКА МЕТОДИКИ ПОБУДОВИ ХМАРНОГО СХОВИЩА НА БАЗІ СУЧАСНИХ МЕТОДІВ ЗАХИСТУ ДАНИХ»

Виконав: студент групи ПДМ-64 Мислюк Андрій Сергійович

Керівник: доктор технічних наук, професор кафедри ІПЗ, Бондарчук Андрій Петрович

Київ - 2024

МЕТА, ОБ'ЄКТА ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: розробка методики побудови хмарного сховища, яка інтегрує сучасні методи захисту даних, з акцентом на розробці ефективних систем авторизації та контролю сесій.

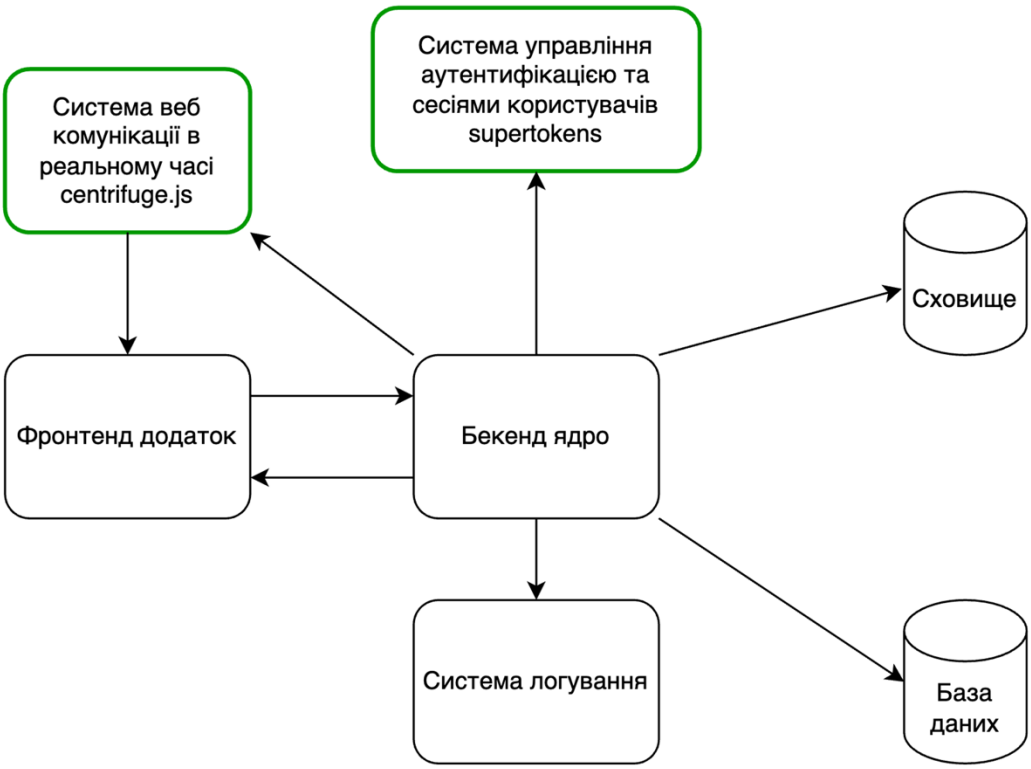
Об'єкт дослідження: побудова та вдосконалення хмарних сховищ з використанням сучасних методів захисту даних.

Предмет дослідження: методи та засоби для створення хмарного сховища, що включає інноваційні підходи в захисті даних і механізми авторизації та контролю сесій.

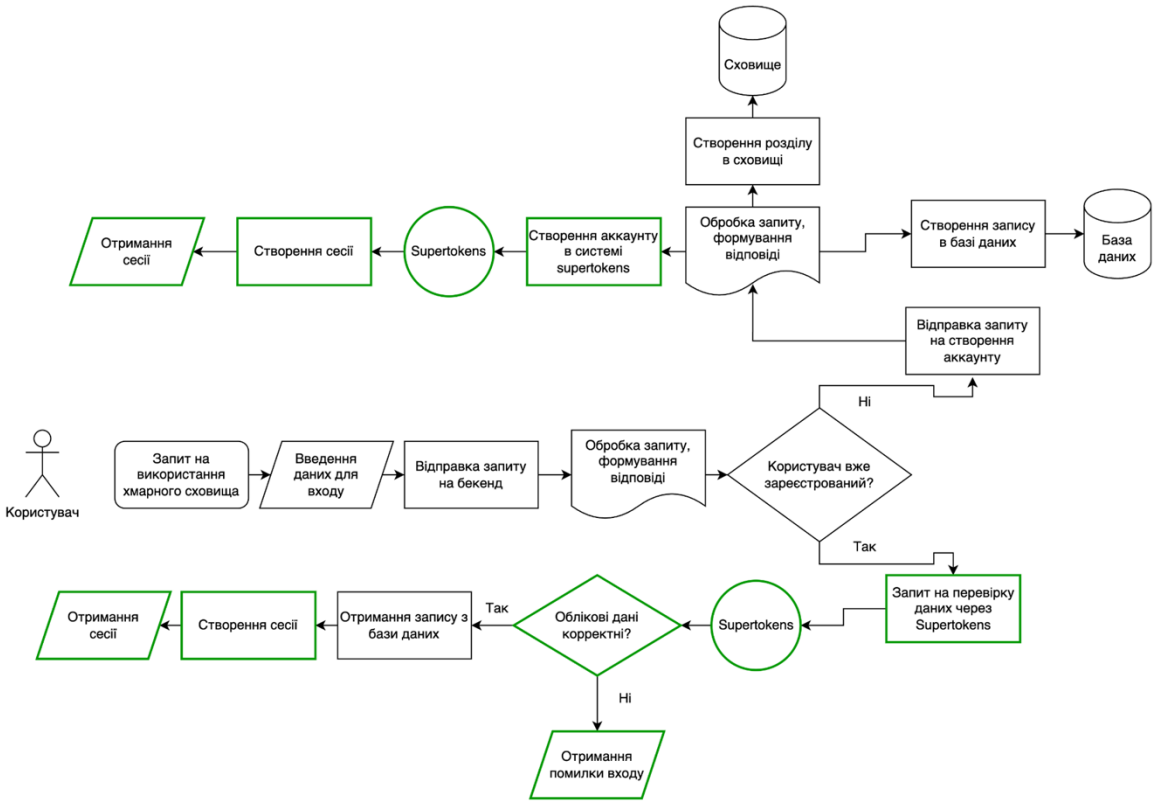
ПОРІВНЯННЯ ІСНУЮЧИХ ХМАРНИХ СХОВИЩ

Особливості	AWS (Amazon Web Services)	Azure (Microsoft)	Google Cloud Platform	IBM Cloud
Особливості архітектури	Децентралізована, з автоматизованими рішеннями для оптимізації роботи	Єдина платформа з тісною інтеграцією з Windows Server, SQL Server, Active Directory	Орієнтована на аналітику, з інтеграцією AI і машинного навчання	Гібридна модель з можливостями інтеграції з приватними хмарними та локальними системами
Масштабованість	Автоматичне горизонтальне та вертикальне масштабування, підтримка великих обсягів даних	Гнучкість у масштабуванні, з акцентом на підтримку корпоративних додатків	Швидка адаптація до змін навантаження, ефективне управління великими даними	Сильні можливості для масштабування у межах гібридної інфраструктури
Засоби безпеки	Розширене шифрування, Identity and Access Management	Інтегрована безпека з Microsoft продуктами, Azure Security Center	Сильне шифрування даних, Google's Secure-by-design інфраструктура	Власні рішення для кібербезпеки, включаючи шифрування та управління ідентифікацією

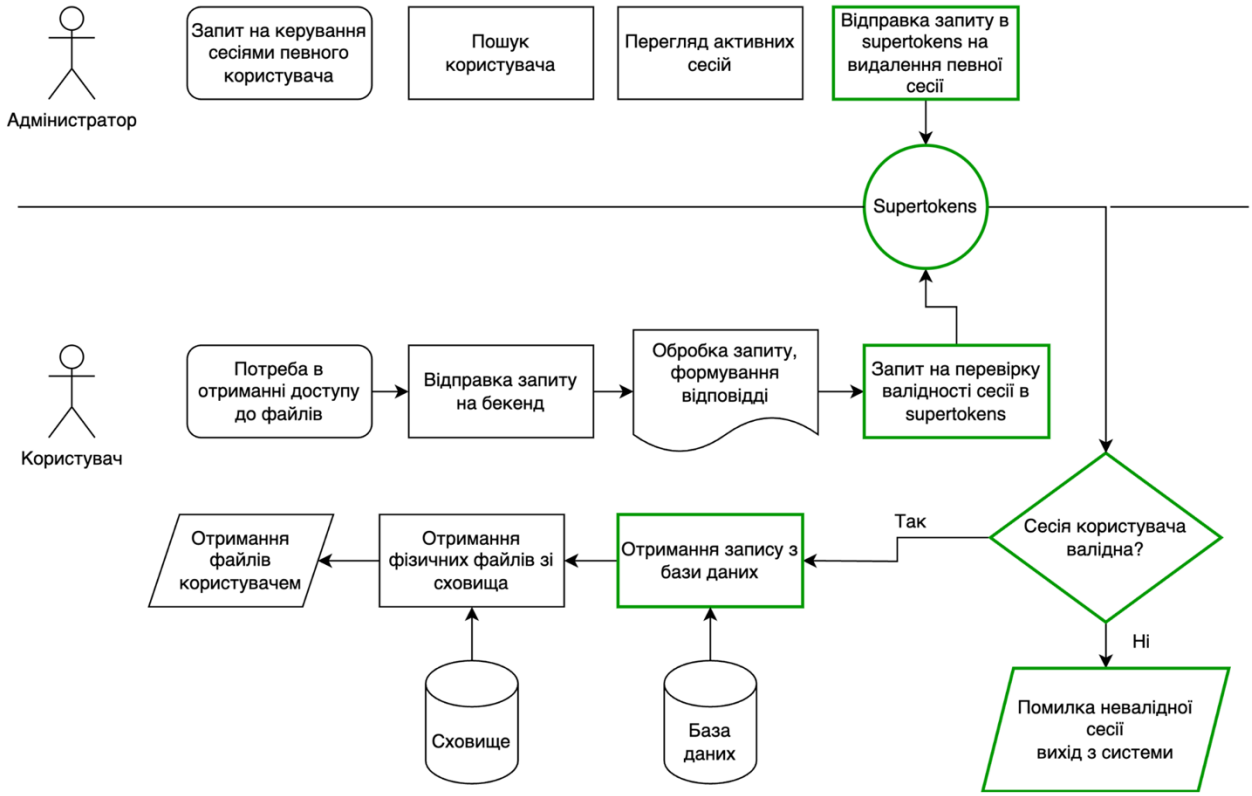
АРХІТЕКТУРНА ДІАГРАМА ХМАРНОГО СХОВИЩА



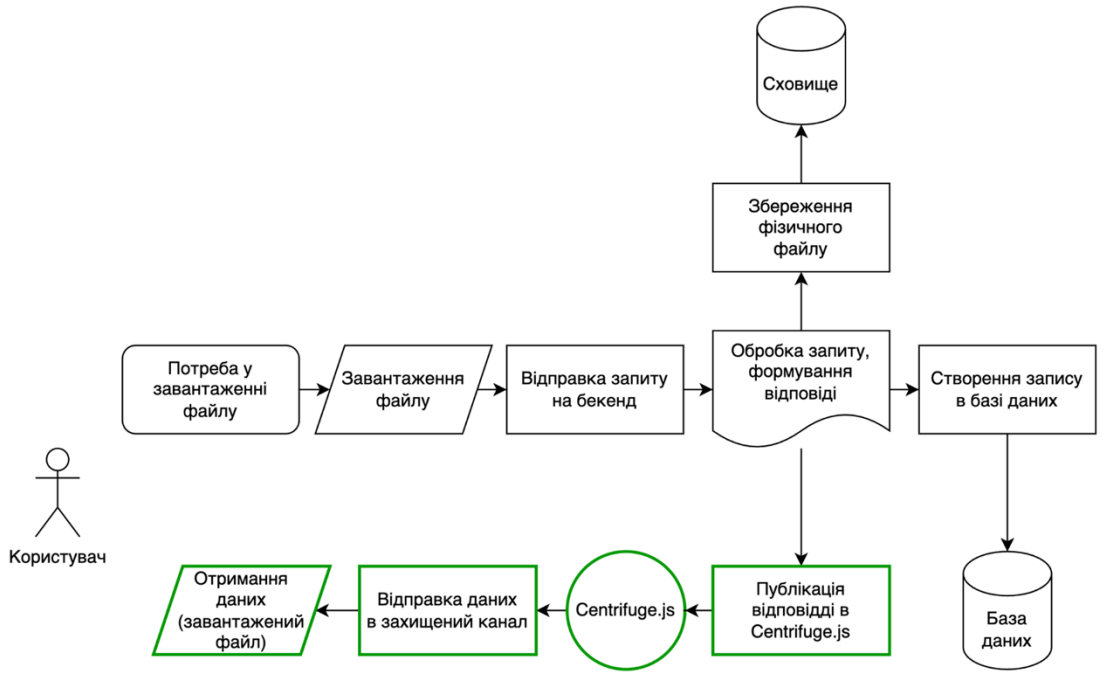
АЛГОРИТМ ПРОЦЕСУ АВТОРИЗАЦІЇ КОРИСТУВАЧА



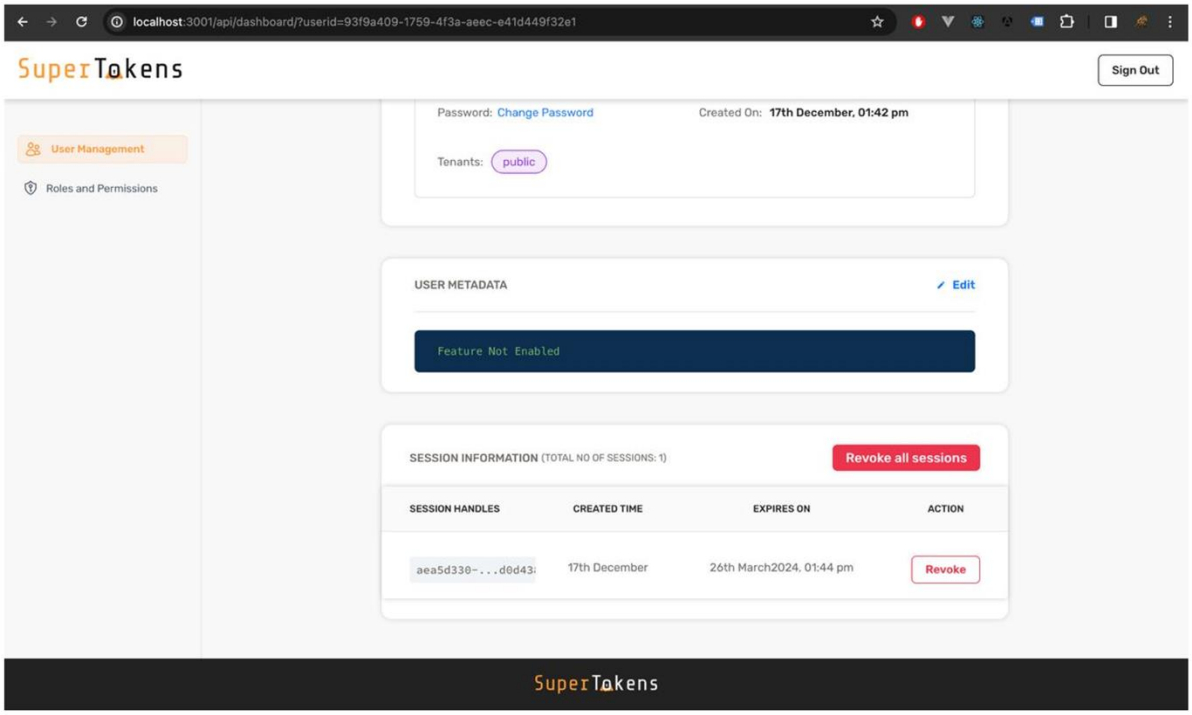
АЛГОРИТМ ПРОЦЕСУ КЕРУВАННЯ СЕСІЯМИ КОРИСТУВАЧА



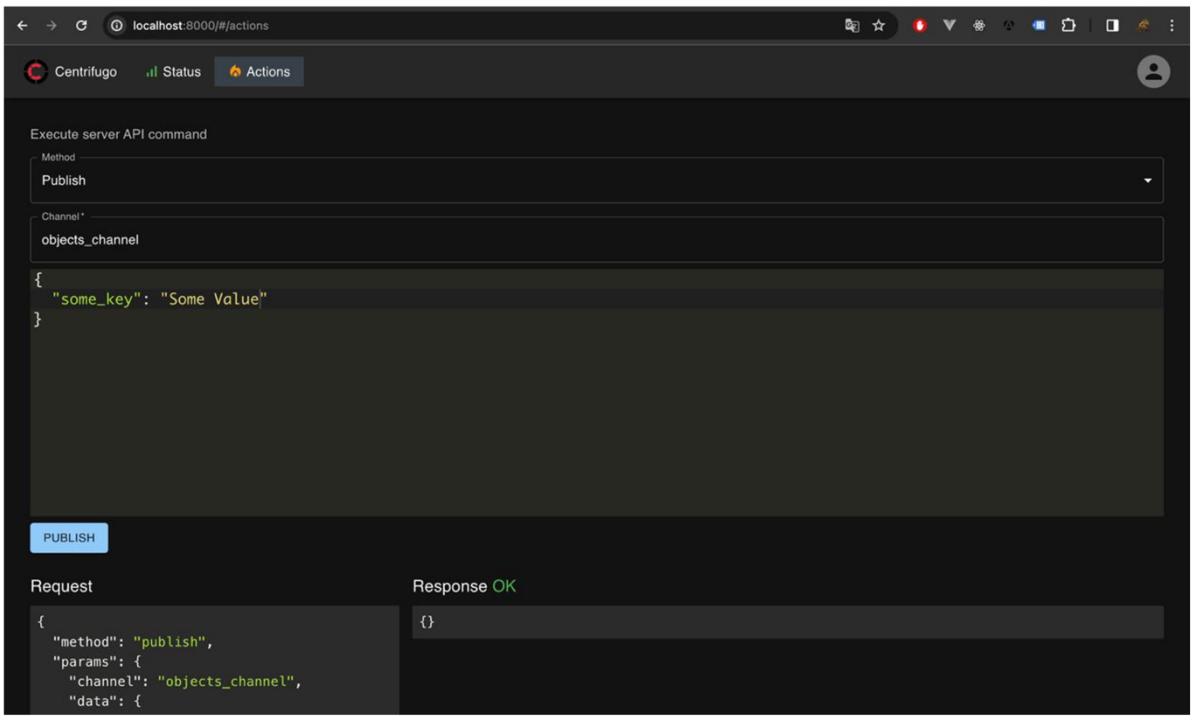
АЛГОРИТМ ПРОЦЕСУ ОТРИМАННЯ ДАНИХ ПО ЗАХИЩЕННОМУ КАНАЛУ



ПРИКЛАД КЕРУВАННЯ АКТИВНИМИ СЕСІЯМИ КОРИСТУВАЧІВ



ПРИКЛАД ВІДПРАВКИ ПОВІДОМЛЕНЬ ЧЕРЕЗ CENTRIFUGO



МАТЕМАТИЧНІ ФОРМУЛИ ДЛЯ ОЦІНКИ ЕФЕКТИВНОСТІ СИСТЕМИ

Формула часу відгуку (L):

$L_{\text{старий}} = L_{\text{базовий}} + D$: Перед впровадженням змін загальний час відгуку системи складається з базового часу реакції та додаткової затримки через неоптимізовані процеси.
$L_{\text{новий}} = L_{\text{базовий}} + \frac{D}{4}$: Після оптимізації системи додаткова затримка зменшується, що призводить до загального покращення часу відгуку.

Формула пропускної здатності (P):

$P = K \times M$: Пропускна здатність є продуктом коефіцієнта ефективності системи та її теоретично максимальної пропускної здатності.
K змінився з $\frac{1}{30}$ на $\frac{1}{5}$: Відображає збільшення ефективності системи, що дозволяє обробляти більше запитів за одиницю часу.

Формула Mean Time Between Failures (MTBF):

$MTBF = \frac{\text{загальний час роботи системи}}{\text{кількість збоїв}}$: Вказує на середній час між збоями системи, де збільшення $MTBF$ свідчить про підвищення надійності та стабільності системи.

Формула витрат на підтримку (V):

$V_{\text{після}} = V_{\text{до}} \times E$: Показує, як впровадження покращень в систему впливає на зниження витрат на її підтримку, де E є коефіцієнтом ефективності цих покращень.

ОЦІНКА МАСШТАБОВАНОСТІ ТА НАДІЙНОСТІ СИСТЕМИ

Показник	До впровадження методів захисту	Після впровадження методів захисту
Час відгуку системи (мс)	500	100
Пропускна здатність під час пікових навантажень (запитів/сек)	50	300
Стабільність під час пікових навантажень	часткові збої	стабільна
Витрати на підтримку	високі	нижчі
Технології та протоколи аутентифікації та авторизації	Cookies і Сесії, Basic Authentication	JWT, OAuth 2.0, OpenID Connect
Рівень масштабованості системи	Низький	високий

ВИСНОВКИ

1. В результаті аналізу архітектури хмарного сховища було виявлено, що сучасні хмарні сховища надають ефективне, гнучке, та масштабоване зберігання даних. Особлива увага була приділена складовим елементам архітектури, таким як сервери даних, системи управління даними та мережеві інфраструктури, що забезпечують оптимальну працездатність та безпеку даних.
2. Дослідження показало, що процедура реєстрації та аутентифікації в хмарних сховищах є ключовою для забезпечення безпеки даних. Багаторівнева аутентифікація та використання сучасних методів верифікації користувачів значно знижують ризик несанкціонованого доступу до даних.
3. В роботі було розроблено нові стратегії шифрування та методи захисту даних, які підвищують рівень безпеки в хмарних сховищах. Ці методи включають в себе передові техніки шифрування, які забезпечують цілісність та конфіденційність даних.
4. Оцінка показала, що хмарні сховища здатні ефективно масштабуватися для відповіді на зростаючі потреби в обробці та зберіганні даних. Надійність системи забезпечується через використання резервного копіювання, відновлення даних та різноманітні механізми відновлення після збоїв.
5. Практична реалізація теоретичних концепцій у вигляді демонстраційного додатку дозволила продемонструвати ефективність запропонованих рішень. Це підтвердило важливість інтеграції теоретичних знань з практичним застосуванням в реальних умовах.

ПУБЛІКАЦІЇ ТА АПРОБАЦІЯ РОБОТИ

Тези доповідей:

Інтеграція штучного інтелекту для підвищення захисту даних хмарного сховища / Мислюк А.С., Бондарчук А.П. // Науково-практична конференція «Проблеми Комп'ютерної Інженерії», збірник тез. – 1 грудня 2023 року, Державний Університет Інформаційно-Комунікаційних Технологій, Київ, Україна. – С. 32.

Сучасні методи захисту даних у сфері фінтех: інноваційні підходи та їх ефективність / Мислюк А.С., Бондарчук А.П. // VI Міжнародна наукова конференція «Науковий простір: актуальні питання, досягнення та інновації», збірник тез. – 15 грудня 2023 року, м. Київ, Україна. – С. 363.

ДЯКУЮ ЗА УВАГУ!